对话系统中的核心技术

汇报人：扣扣

大纲

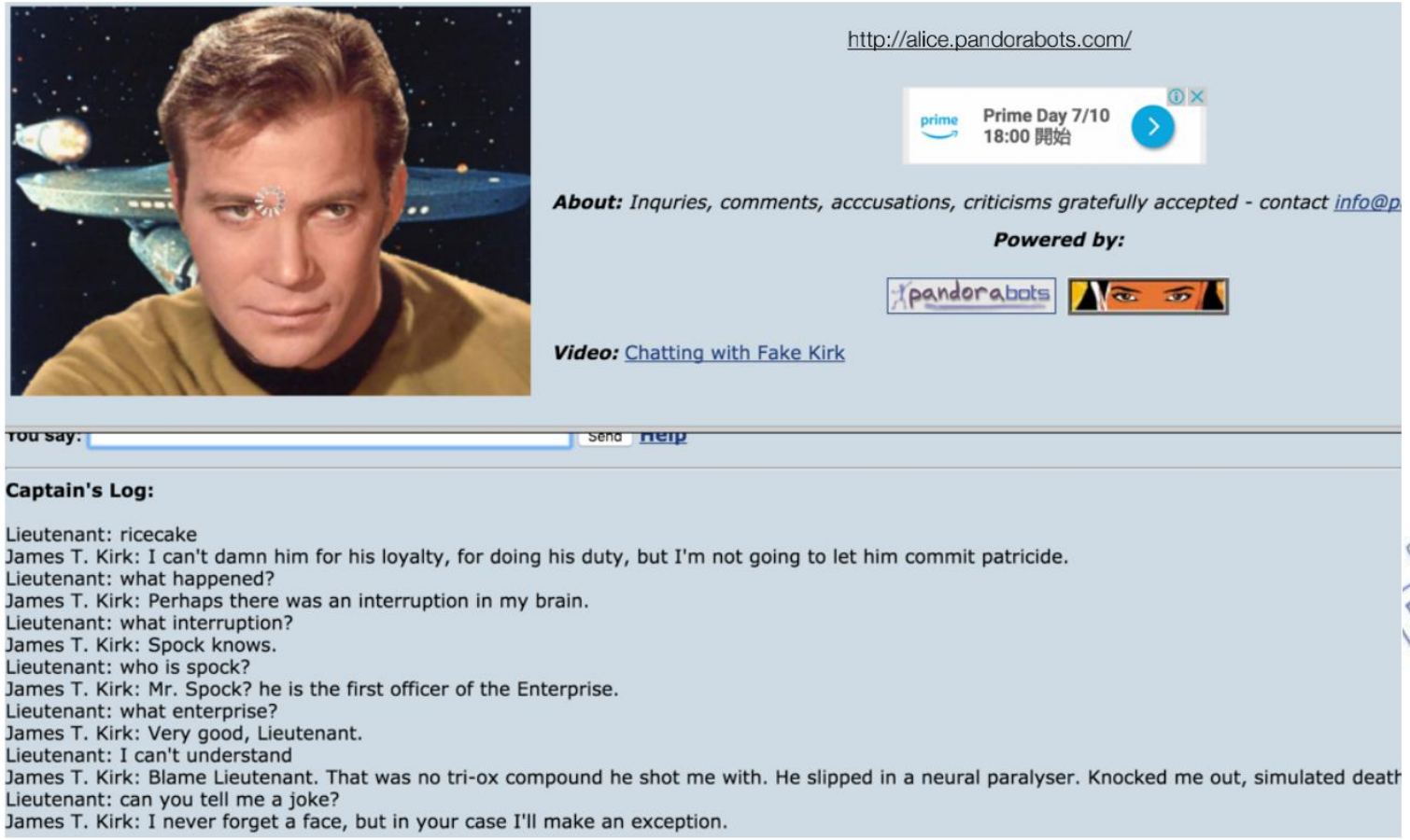
- 对话系统概述
- 闲聊式对话系统的关键技术
- 检索式对话系统的关键技术
- 知识问答式对话系统的关键技术
- 任务式对话系统的关键技术

贪心科技 GREEDY TECHNOLOGY
NeuHub AI Links All

对话系统概述

The World's First Chatbot

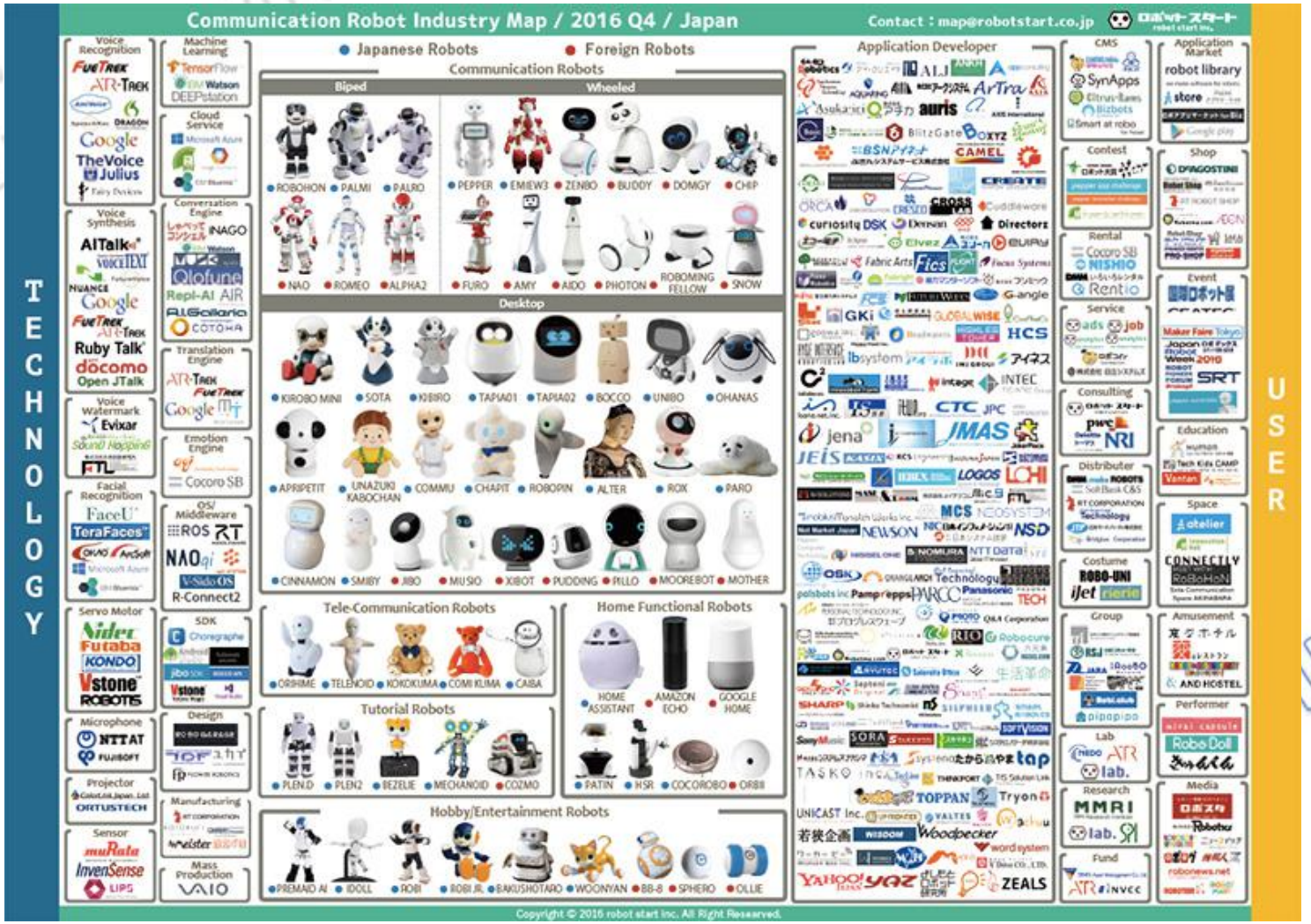
ALIZE


http://alice.pandorabots.com/
Prime Day 7/10
18:00 開始
About: Inquries, comments, acccusations, criticisms gratefully accepted - contact info@p
Powered by:
pandorabots
Video: Chatting with Fake Kirk
You say:
Send
Help
Captain's Log:
Lieutenant: ricecake
James T. Kirk: I can't damn him for his loyalty, for doing his duty, but I'm not going to let him commit patricide.
Lieutenant: what happened?
James T. Kirk: Perhaps there was an interruption in my brain.
Lieutenant: what interruption?
James T. Kirk: Spock knows.
Lieutenant: who is spock?
James T. Kirk: Mr. Spock? he is the first officer of the Enterprise.
Lieutenant: what enterprise?
James T. Kirk: Very good, Lieutenant.
Lieutenant: I can't understand
James T. Kirk: Blame Lieutenant. That was no tri-ox compound he shot me with. He slipped in a neural paralyser. Knocked me out, simulated death
Lieutenant: can you tell me a joke?
James T. Kirk: I never forget a face, but in your case I'll make an exception.

京东智联云 & AI Hub AI Links All

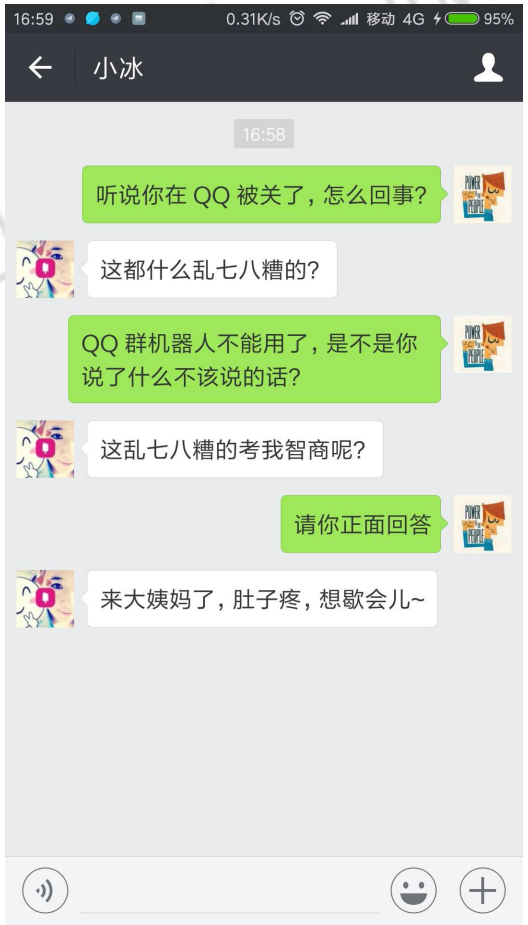
对话系统概述

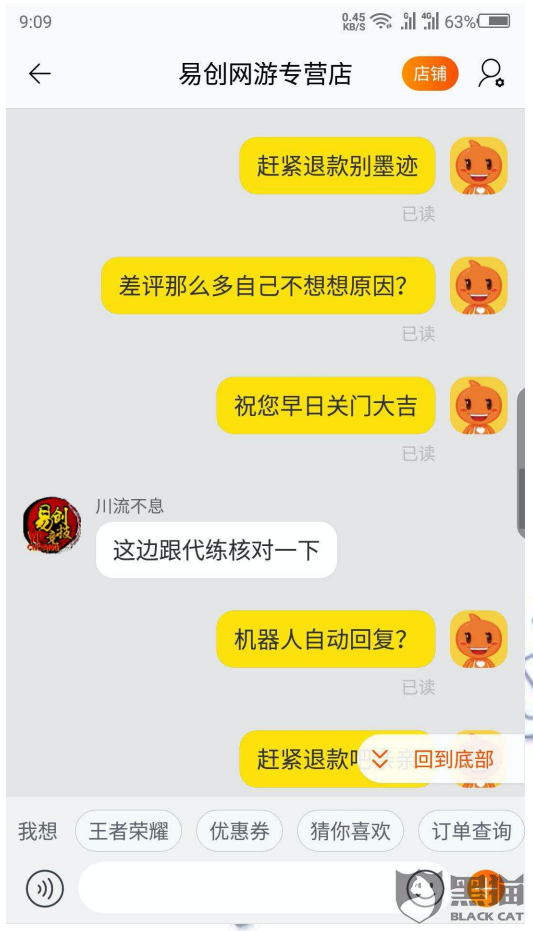
应用广泛
产品众多



贪心科技 | 让每个人享受个性化教育服务

对话系统概述





AI Links All

对话系统概述

我要预约一个取件

请问取件地址是哪里?

海淀区中关村大街1号1802

请问希望什么时间取件？最早的取件时间是今天11:00

可以，那就今天11点

请告诉我寄件人电话，我就能帮您完成预约了

好的，18612345678

已为您预约取件，快递员11:00上门，请耐心等待

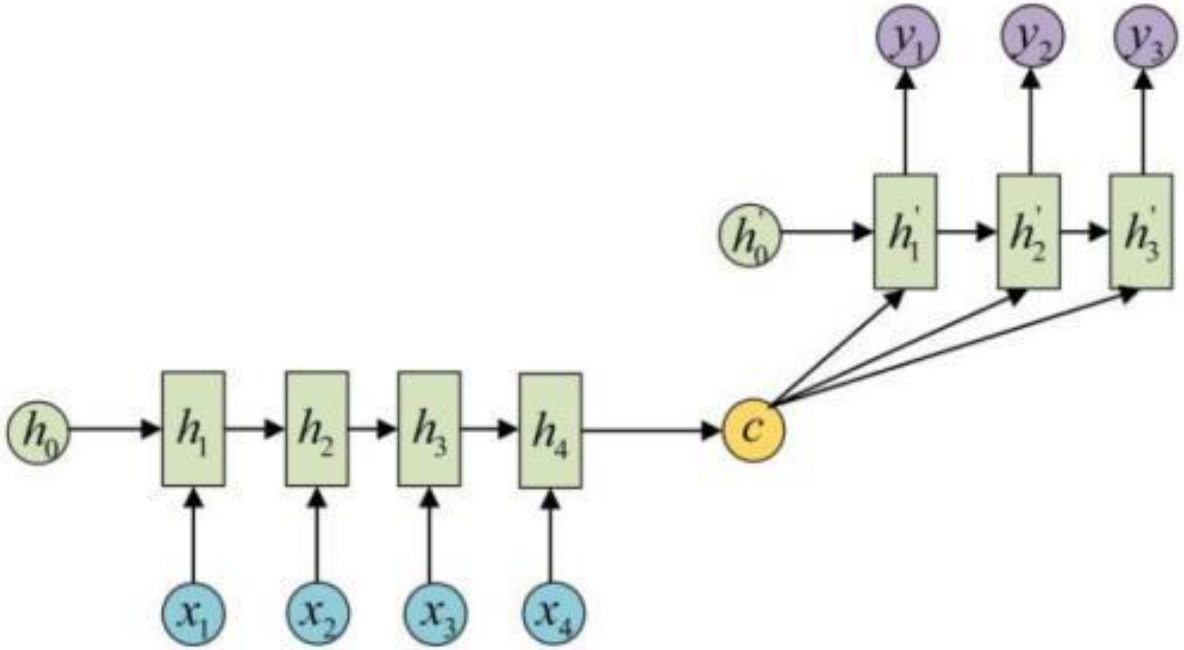
贪心科技 GREEDY TECHNOLOGY

京东智联云 & 贪心科技 GREEDY TECHNOLOGY

闲聊式对话系统的关键技术

NeuHub AI Links All

闲聊式对话系统的关键技术



- 自主动手实现 seq2seq + attention
- 基于现有API（小冰、图灵、腾讯、青云客）
- 基于开源项目（ChatterBot）

NeuHub AI Links All

闲聊式对话系统的关键技术

- ChatterBot:https://chatterbot.readthedocs.io/en/stable/

```
user: Good morning! How are you doing?
bot:  I am doing very well, thank you for asking.
user: You're welcome.
bot:  Do you like hats?
```

NeuHub
AI Links All

闲聊式对话系统的关键技术

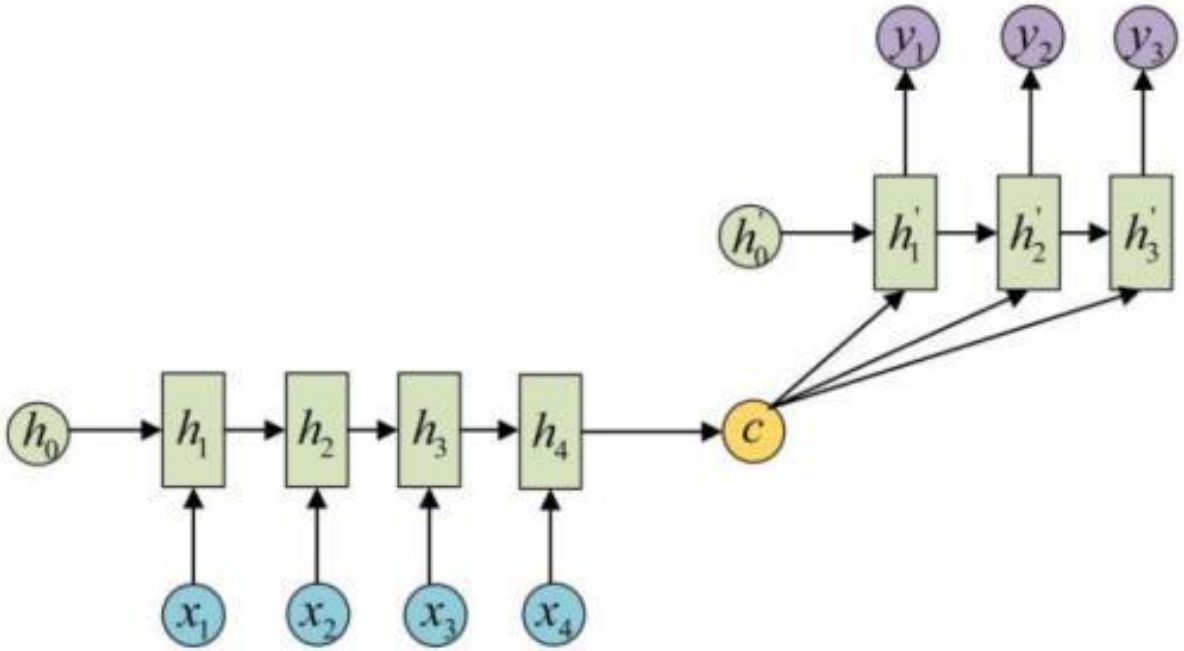
京东智联云 & GREEDY TECHNOLOGY

- ChatterBot:https://chatterbot.readthedocs.io/en/stable/

```python
from chatterbot import ChatBot
chatbot = ChatBot("Coco")
from chatterbot.trainers import ListTrainer

conversation = [
    "Hello",
    "Hi there!",
    "How are you doing?",
    "I'm doing great.",
    "That is good to hear",
    "Thank you.",
    "You're welcome."
]

trainer = ListTrainer(chatbot)

trainer.train(conversation)
response = chatbot.get_response("Good morning!")
print(response)
```

AI Links All

闲聊式对话系统的关键技术



- 生成式对话系统（开放域）的评估难，需要高度依赖人工评定。
- 对话回复与语境高度耦合：显式的聊天历史以及隐式的说话人的属性、性格以及聊天发生时的场合和受众等。

闲聊式对话系统的关键技术

- Emotional Chatting Machine：情绪化聊天机器人

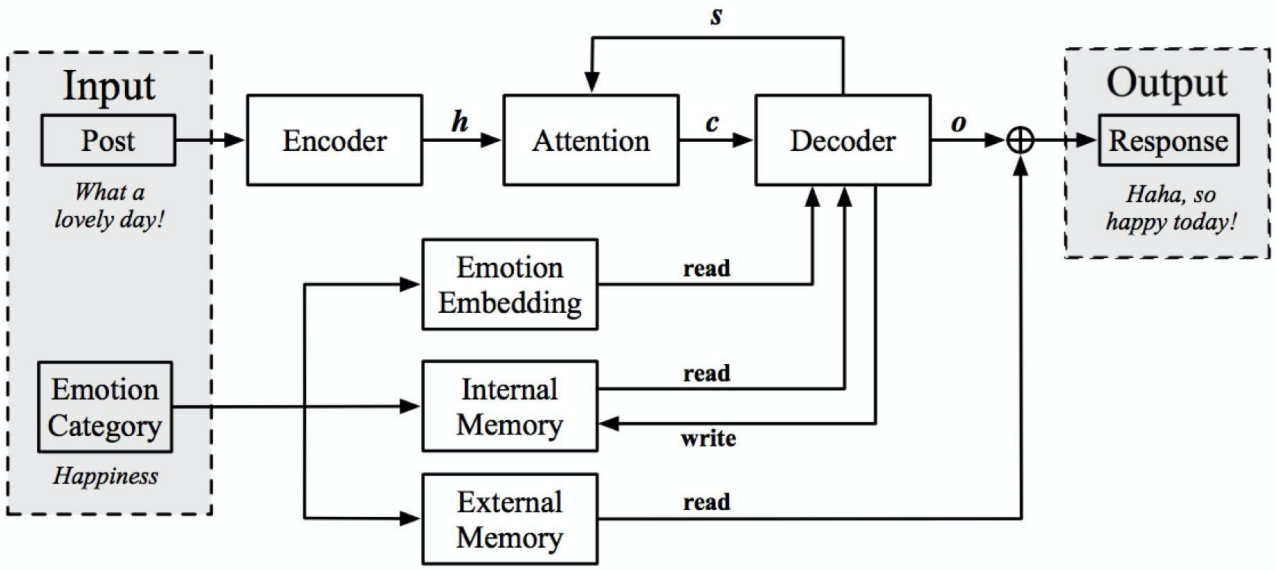
指定情绪	Emotional Chatting Machine
POST	早上的天气不错啊！哈哈哈哈哈哈
喜欢	今天的天气不错啊！
悲伤	天气好，心情不好。
厌恶	这是哪里啊？求解释
愤怒	怎么了？什么情况？
快乐	今天的天气真好啊！

POST	送一个这样的蛋糕，给你美丽的闺蜜吧！
喜欢	好漂亮的蛋糕啊！
悲伤	为什么我觉得很恶心
厌恶	这样的蛋糕你忍心吗？
愤怒	有没有想开网店的？
快乐	哈哈，好可爱的蛋糕！

闲聊式对话系统的关键技术

京东智联云 & GREEDY TECHNOLOGY

- Emotional Chatting Machine：情绪化聊天机器人



用户输入：“What a lovely day!”
Encoder + decoder + attention -> c

指定情感类别：“Happiness”，经过索引得到情感类别嵌入向量，初始的情感状态记忆向量和相应的情感词表。

decoder 接受经过注意力机制的问题向量 c，情感类别嵌入向量和初始的情感状态记忆向量作为输入，通过循环神经网络生成下个词的生成概率 o;
经过情感词表对情感词和非情感词的加权，得到最终词的生成概率，通过采样即可得到输出“Haha, so happy today!”。

闲聊式对话系统的关键技术

京东智联云 & AI

控制性地生成对话相关论文：

Zhao, Tiancheng, Ran Zhao, and Maxine Eskenazi. “Learning discourse-level diversity for neural dialog models using conditional variational autoencoders.” arXiv preprint arXiv:1703.10960 (2017).

Zhou, Xianda, and William Yang Wang. “Mojitalk: Generating emotional responses at scale.” arXiv preprint arXiv:1711.04090 (2017).

Qian, Qiao, Minlie Huang, Haizhou Zhao, Jingfang Xu, and Xiaoyan Zhu. “Assigning personality/identity to a chatting machine for coherent conversation generation.” arXiv preprint arXiv:1706.02861 (2017).

Zhang, Saizheng, Emily Dinan, Jack Urbanek, Arthur Szlam, Douwe Kiela, and Jason Weston. “Personalizing Dialogue Agents: I have a dog, do you have pets too?.” arXiv preprint arXiv:1801.07243 (2018).

NeuHub AI Links All

贪心科技 GREEDY TECHNOLOGY

京东智联云 & 贪心科技

检索式对话系统的关键技术

NeuHub AI Links All

例：搭建一个智能客服系统

常见的问题（FAQ）：

1. 本课程是线上课程还是线下课程？

   回答：线上课程为主

2. 课程有助教吗

   回答： 每门课程都配备专业助教

3. 学习周期是多久啊？

   回答：通常来讲在3-4个月不等

4. 如果不满意可以退款吗？

   回答： 前两周提供无条件退款

5. 老师都是什么背景啊？

   回答： 绝大部分都是全美前10学校的博士

6. 课程会有考试吗

   回答： 有的。一般包括期中和期末

7. 我只有编程基础，可以报名吗

   回答： 对于初级的项目班只要求编程基础

8. 课程有实操吗

   回答：大部分都是实操，动手能力是最重要的

9. 课程为什么贵？

   回答：跟别的知识付费不一样，我们会提供很多教学服务，辅助完成学员做完所有的项目

10. 课程学完了能做什么？

    回答： 可以找相关岗位的工作问题不大

11. 课程多久开一次啊？

    回答： 我们每个月开一期，但价格通常会不断升高

NeuHub AI Links All

例：搭建一个智能客服系统

京东智联云 &

常见的问题（FAQ）：

1. 本课程是线上课程还是线下课程？

回答： 本课程是线上课程还是线下课程？

2. 课程有助教吗

回答： 每门课程都配备专业助教

3. 学习周期是多久啊？

回答：通常来讲在3-4个月不等

4. 如果不满意可以退款吗？

回答： 前两周提供无条件退款

5. 老师都是什么背景啊？

回答： 绝大部分都是全美前10学校的博士

6. 课程会有考试吗

回答： 有的。一般包括期中和期末

7. 我只有编程基础，可以报名吗

回答：对于初级的项目班只要求编程基础

8. 课程有实操吗

回答：大部分都是实操，动手能力是最重要的

9. 课程为什么贵？

回答：跟别的知识付费不一样，我们会提供很多教学服务，辅助完成学员做完所有的项目

10. 课程学完了能做什么？

回答： 可以找相关岗位的工作问题不大

11. 下次期班是什么时候？

回答： 我们每个月开一期，但价格通常会不断升高

可不可以用这些语料库来搭建一个智能客服系统？

NeuHub AI Links All

例：搭建一个智能客服系统

&

常见的问题（FAQ）：

1. 本课程是线上课程还是线下课程？

回答： 本课程是线上课程还是线下课程？

2. 课程有助教吗

回答： 每门课程都配备专业助教

3. 学习周期是多久啊？

回答：通常来讲在3-4个月不等

4. 如果不满意可以退款吗？

回答： 前两周提供无条件退款

5. 老师都是什么背景啊？

回答： 绝大部分都是名校硕士及博士

6. 课程会有考试吗

回答： 有的。一般包括期中和期末

7. 我只有编程基础，可以报名吗

回答： 对于初级的项目班只要求编程基础

8. 课程有实操吗

回答：大部分都是实操，动手能力是最重要的

9. 课程为什么贵？

回答：跟别的知识付费不一样，我们会提供很多教学服务，辅助完成学员做完所有的项目

10. 课程学完了能做什么？

回答： 可以找相关岗位的工作问题不大

11. 下次期班是什么时候？

回答： 我们每个月开一期，但价格通常会不断升高

相似度：0.1

相似度：0.05

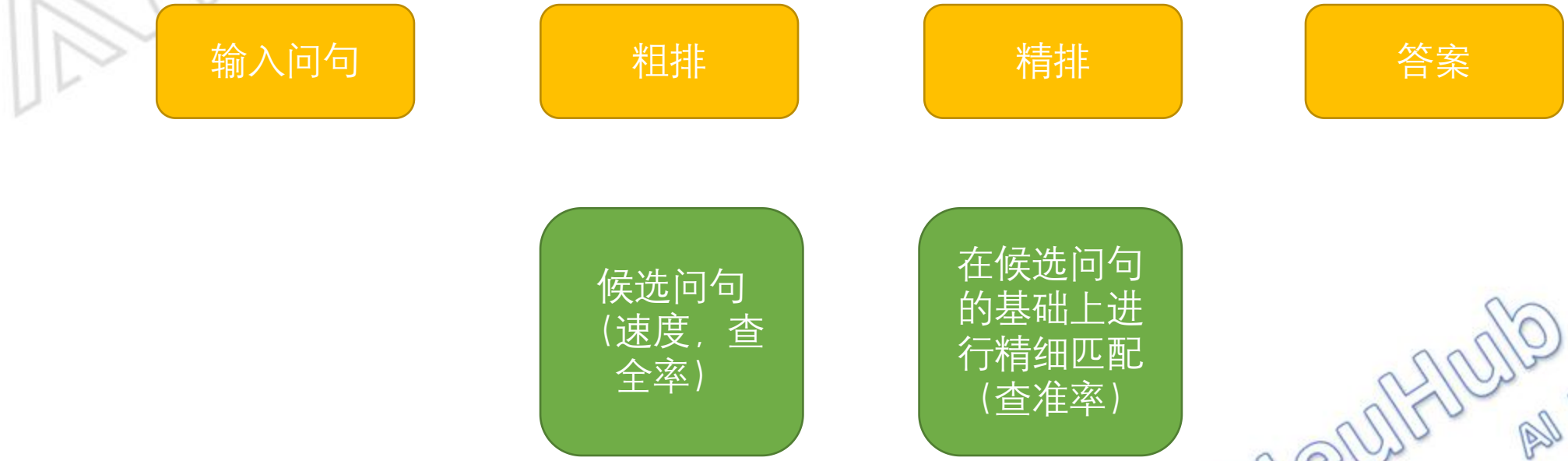
相似度：0.9

用户输入：“我想了解老师的背景”

机器回复

NeuHub AI Links All

检索式对话系统的关键技术



NeuHub
AI Links All

检索式对话系统的关键技术

&

- 如何表征文本
- 如何计算文本间距离：WMD
- 如何快速计算

NeuHub
AI Links All

文本的简单表示：基于频次的词袋模型

Bag of words (BoW)

Very good drama although it appeared to have a few blank areas leaving the viewers to fill in the action for themselves. I can imagine life being this way for someone who can neither read nor write. This film simply smacked of the real world: the wife who is suddenly the sole supporter, the live-in relatives and their quarrels, the troubled child who gets knocked up and then, typically, drops out of school, a jackass husband who takes the nest egg and buys beer with it. 2 thumbs up… very very very good movie.

('the', 8),
(',', 5),
('very', 4),
('.', 4),
('who', 4),
('and', 3),
('good', 2),
('it', 2),
('to', 2),
('a', 2),
('for', 2),
('can', 2),
('this', 2),
('of', 2),
('drama', 1),
('although', 1),
('appeared', 1),
('have', 1),
('few', 1),
('blank', 1)
…..

词袋模型（Bag-of-Words, BOW），顾名思义，即把文本中的词用袋子装起来统一作为文本的表示。

基于词频的词袋模型是将文本进行数值化表示的一种简单模型。

NeuHub
AI Links All

文本的简单表示：基于频次的词袋模型

Bag of words (BoW)

Very good drama although it appeared to have a few blank areas leaving the viewers to fill in the action for themselves. I can imagine life being this way for someone who can neither read nor write. This film simply smacked of the real world: the wife who is suddenly the sole supporter, the live-in relatives and their quarrels, the troubled child who gets knocked up and then, typically, drops out of school, a jackass husband who takes the nest egg and buys beer with it. 2 thumbs up… very very very good movie.

('the', 8),
(',', 5),
('very', 4),
('.', 4),
('who', 4),
('and', 3),
('good', 2),
('it', 2),
('to', 2),
('a', 2),
('for', 2),
('can', 2),
('this', 2),
('of', 2),
('drama', 1),
('although', 1),
('appeared', 1),
('have', 1),
('few', 1),
('blank', 1)
…..

假设数据集为两个经过预处理且分好词的句子：

1）你/站在/桥上/看/风景/看/风景/的/人/在/楼上/看/你

2）明月/装饰/了/你/的/窗子/你/装饰/了/别人/的/梦

可得对应的长度为15的词典如下：

[“你”，“站在”，“桥上”，“看”，“风景”，“的”，“人”，“在”，“楼上”，“明月”，“装饰”，“了”，“窗子”，“别人”，“梦”]

NeuHub AI Links U

文本的简单表示：基于频次的词袋模型

假设数据集为两个经过预处理且分好词的句子：
1）你/站在/桥上/看/风景/看/风景/的/人/在/楼上/看/你
2）明月/装饰/了/你/的/窗子/你/装饰/了/别人/的/梦

可得对应的长度为15的词典如下：
[“你”，“站在”，“桥上”，“看”，“风景”，“的”，“人”，“在”，“楼上”，“明月”，“装饰”，“了”，“窗子”，“别人”，“梦”]

以上两个句子的向量化表示如下：
1）[2, 1, 1, 3, 2, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0]
2）[2, 0, 0, 0, 0, 2, 0, 0, 0, 1, 2, 2, 1, 1, 1]

优点：
简单易用，原理明晰；

缺点：
没有考虑词序、词之间的联系以及文法，
丢失了许多重要信息。

NeuHub AI Links All

文本的简单表示：基于TF-IDF

TF-IDF（Term Frequency-Inverse Document Frequency）的核心包含两部分：

- TF表示某个词在某一文本中出现的频率。
- IDF为逆向文档频率，与某词在综合语料库中出现的频率相关。

TF-IDF可用于评估某一词对所在文本的重要程度，其在当前文本中出现频次越多，而在其它文本中出现频次越少，可认为此词越是重要。

NeuHub AI Links All

文本的简单表示：基于TF-IDF

京东智联云 & 贪心科技

$$\text{词频(TF)} = \text{某个词在文章中的出现次数}$$

$$\text{词频(TF)} = \frac{\text{某个词在文章中的出现次数}}{\text{文章的总词数}}$$

考虑到文章有长短之分，为了便于不同文章的比较，进行"词频"标准化。

NeuralAI Links All

文本的简单表示：基于TF-IDF

- 需要一个语料库用来模拟语言的使用环境。如果一个词越常见，那么分母就越大，逆文档频率就越小越接近0。
- 分母之所以要加1，是为了避免分母为0（即所有文档都不包含该词）。log表示对得到的值取对数。

NeuHub AI links All

文本的简单表示：基于TF-IDF

&

$$\mathrm{TF-IDF} = \text{词频}(\mathrm{TF}) \times \text{逆文档频率}(\mathrm{IDF})$$

TF-IDF与一个词在文档中的出现次数成正比，与该词在整个语言中的出现次数成反比。

NeuHub
AI Links All

文本的简单表示：基于TF-IDF

京东智联云 &

$$\mathrm{TF-IDF} = \text{词频}(\mathrm{TF}) \times \text{逆文档频率}(\mathrm{IDF})$$

优缺点

- 优点：简单快速，易于理解
- 缺点：只用词频来衡量文章中的一个词的重要性不够全面，有时候重要的词出现的可能并不多，而且这种计算无法体现位置信息，无法体现词在上下文中的重要性。

NeuHub AI Links All

句向量:预训练词向量求平均

&

优点：计算简便
缺点：无法考虑到词汇重要程度的差异（简单的例子VS简单例子）；未考虑语序（人生如戏VS戏如人生）

中文词向量链接：https://github.com/Embedding/Chinese-Word-Vectors

NeuHub
AI Links All

句向量:加权平均

TFIDF作为单词权重
优点：考虑词汇不同的重要程度

NeuHub
AI Links All

句向量:加权平均

https://openreview.net/pdf?id=SyK00v5xx
https://github.com/PrincetonML/SIF

SIF 加权平均
1） 对句子中的每个词向量，乘以一个权重 a/(a+p_w)，其中 a 是一个常数（原文取 0.0001），p_w 为该词的词频；对于出现频率越高的词，其权重越小；
2） 计算句向量矩阵的第一个主成分 u，让每个句向量减去它在 u （单位向量）上的矢投影；

Algorithm 1 Sentence Embedding

Input: Word embeddings $\{v_w : w \in \mathcal{V}\}$, a set of sentences $\mathcal{S}$, parameter a and estimated probabilities $\{p(w) : w \in \mathcal{V}\}$ of the words.
Output: Sentence embeddings $\{v_s : s \in \mathcal{S}\}$
1: **for all** sentence s in $\mathcal{S}$ **do**
2: $\quad v_s \leftarrow \frac{1}{|s|} \sum_{w \in s} \frac{a}{a+p(w)} v_w$
3: **end for**
4: Form a matrix X whose columns are $\{v_s : s \in \mathcal{S}\}$, and let u be its first singular vector
5: **for all** sentence s in $\mathcal{S}$ **do**
6: $\quad v_s \leftarrow v_s - uu^\top v_s$
7: **end for**

移除所有句子的共有信息，因此保留下来的句子向量更能够表示本身并与其它句子向量产生的差距。

句向量:加权平均

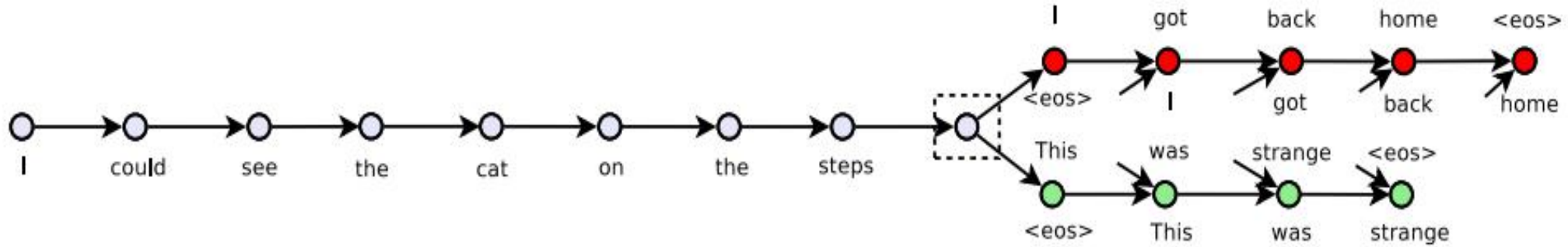
- 基于幂均值（Power Mean）的句向量表示方法。对于由各词向量x组成的一个句子s，计算其P个幂均值。
- x为词向量，其中n为句长，p也可以随意选择。P个幂均值向量进行简单拼接便可作为整个句子的句向量。

$$\left(\frac{x_1^p+\ldots+x^p{}_n}{n}\right)^{1/p}$$

NeuHub AI Links All

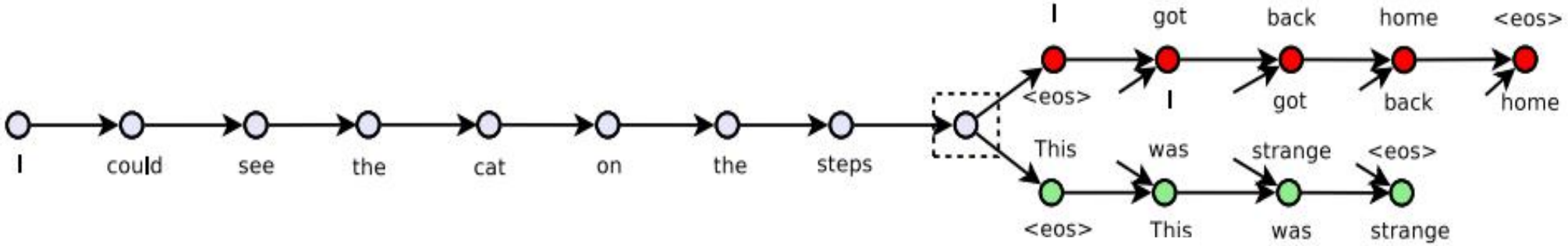
句向量: Skip-Thought Vector

- 类似 Word2Vec语言模型的训练思想，Sentence Embedding 作为模型的副产品。
- 给定一个三元组 s_{i-1}, s_i, s_{i+1} 表示 3 个连续的句子。
- 模型使用 Encoder-Decoder 框架。
- 训练时，由 Encoder 对 s_i 进行编码，然后分别使用两个 Decoder 生成前一句 s_{i-1} 和下一句 s_{i+1}。



贪心科技 | 让每个人享受个性化教育服务

句向量: Skip-Thought Vector



- **Encoder** (GRU)

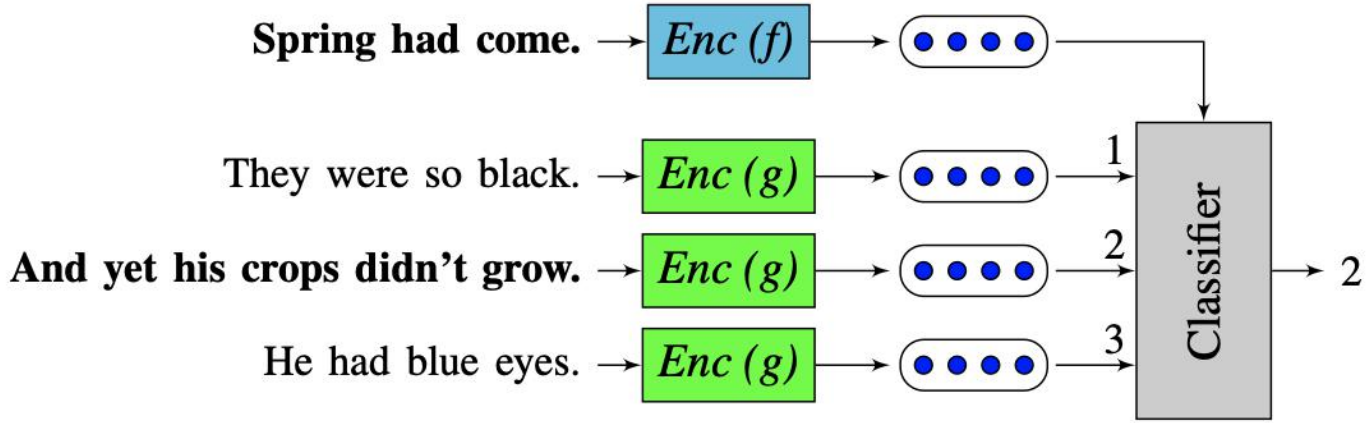
$$
\begin{aligned}
\mathbf{r}^t &= \sigma(\mathbf{W}_r\mathbf{x}^t + \mathbf{U}_r\mathbf{h}^{t-1}) \\
\mathbf{z}^t &= \sigma(\mathbf{W}_z\mathbf{x}^t + \mathbf{U}_z\mathbf{h}^{t-1}) \\
\bar{\mathbf{h}}^t &= \tanh(\mathbf{W}\mathbf{x}^t + \mathbf{U}(\mathbf{r}^t \odot \mathbf{h}^{t-1})) \\
\mathbf{h}^t &= (1 - \mathbf{z}^t) \odot \mathbf{h}^{t-1} + \mathbf{z}^t \odot \bar{\mathbf{h}}^t
\end{aligned}
$$

- **Decoder** (带窥孔的 GRU)

$$
\begin{aligned}
\mathbf{r}^t &= \sigma(\mathbf{W}_r^d\mathbf{x}^{t-1} + \mathbf{U}_r^d\mathbf{h}^{t-1} + \mathbf{C}_r\mathbf{h}_i) \\
\mathbf{z}^t &= \sigma(\mathbf{W}_z^d\mathbf{x}^{t-1} + \mathbf{U}_z^d\mathbf{h}^{t-1} + \mathbf{C}_z\mathbf{h}_i) \\
\bar{\mathbf{h}}^t &= \tanh(\mathbf{W}^d\mathbf{x}^{t-1} + \mathbf{U}^d(\mathbf{r}^t \odot \mathbf{h}^{t-1}) + \mathbf{C}\mathbf{h}_i) \\
\mathbf{h}_{i+1}^t &= (1 - \mathbf{z}^t) \odot \mathbf{h}^{t-1} + \mathbf{z}^t \odot \bar{\mathbf{h}}^t
\end{aligned}
$$

NeuHub AI Links All

句向量: Quick-Thought Vectors

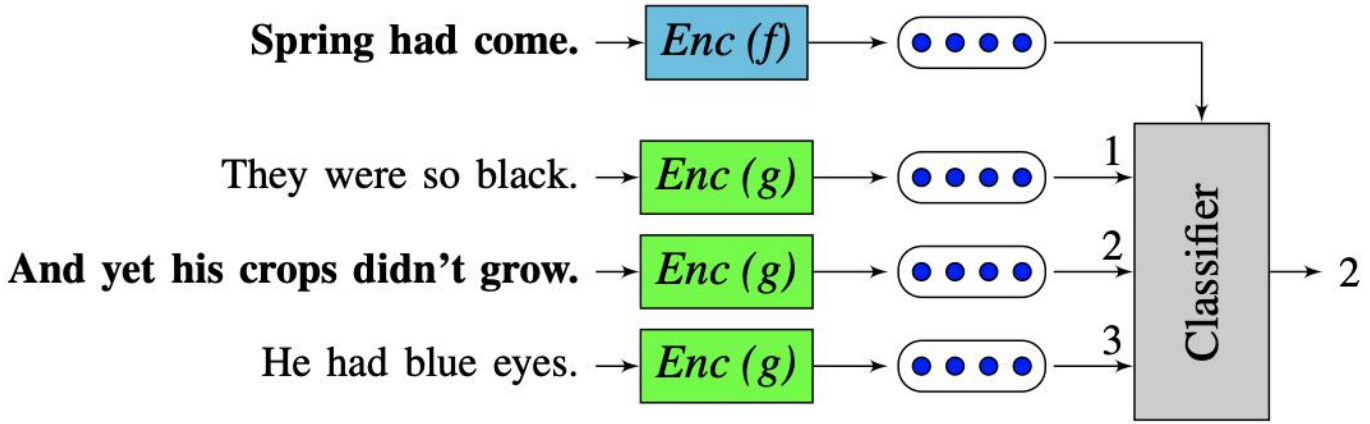


(b) Proposed approach

- Quick-Thoughts模型是对Skip-Thoughts的优化，其主要优点在于训练速度的提升。Skip-Thoughts的结构实质上为生成模型，而且需要训练3个循环神经网络。Quick-Thoughts则将整个过程当作分类任务来处理。
- f,g分别为两个encoder。

NeuHub AI Links All

句向量: Quick-Thought Vectors



(b) Proposed approach

- Scand包含一个上下文句子和许多非上下文句子。
- 目标函数是希望是上下文的句子对所对应的向量积越大越好，反之则越小越好。
- Quick-Thoughts训练完成之后，对于下游任务中的任一句子s，通过f(s)及g(s)计算得的向量拼接而得句向量。

$$p(s_{\text{cand}}|s, S_{\text{cand}}) = \frac{\exp[c(f(s), g(s_{\text{cand}}))]}{\sum_{s' \in S_{\text{cand}}} \exp[c(f(s), g(s'))]}$$

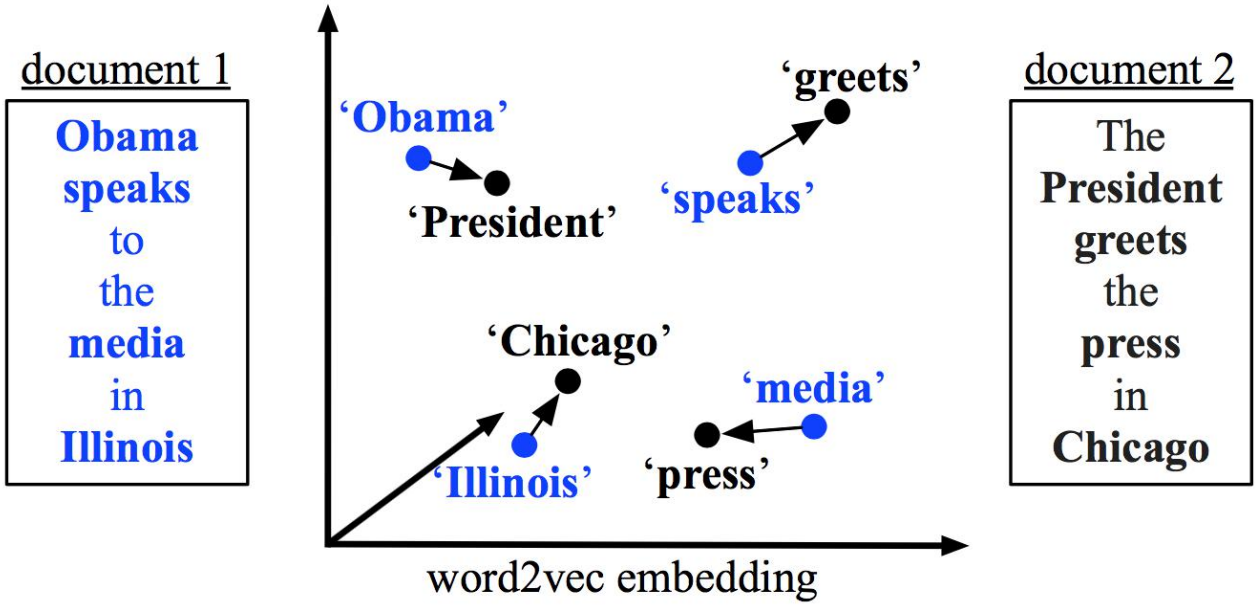
检索式对话系统的关键技术

&

- 如何计算文本间距离：WMD

NeuHub
AI Links All

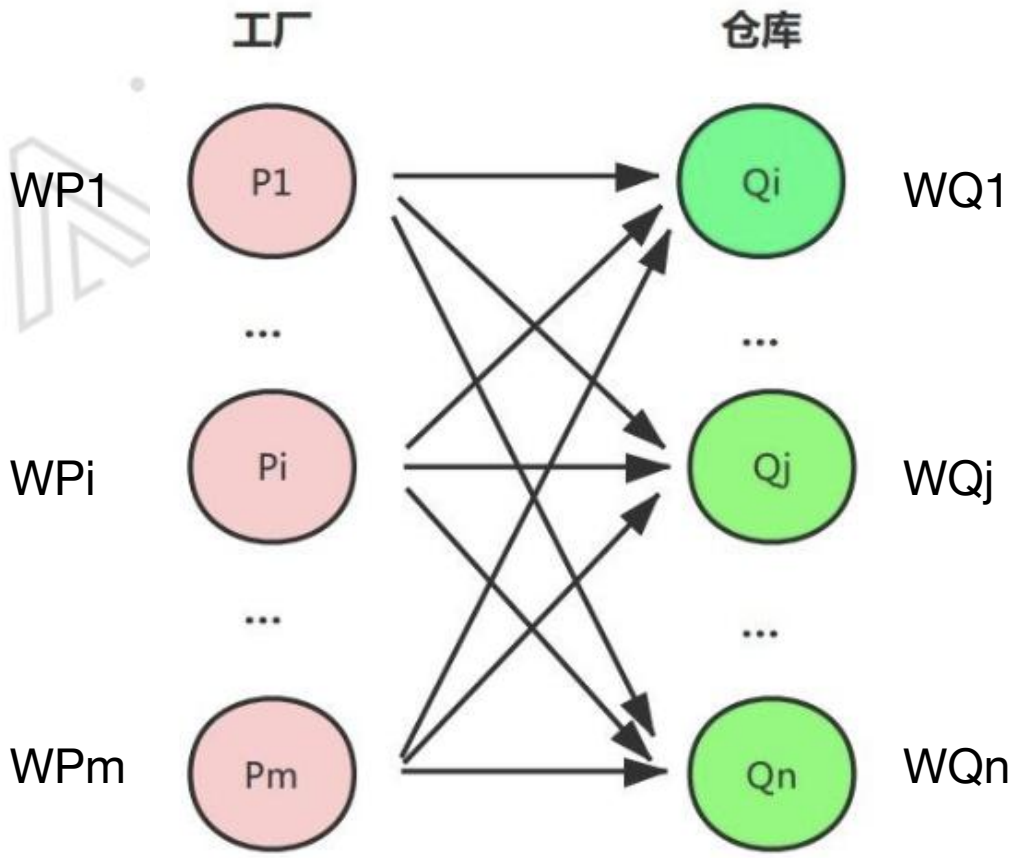
文本相似度计算：WMD



- WMD(Word Mover's Distance）中文称作词移距离。
- 通过寻找两个文本之间所有词之间最小距离之和的配对来度量文本的语义相似度。
- 是EMD(Earth Mover's Distance)在NLP领域的延伸。

NeuHub AI Links All

EMD

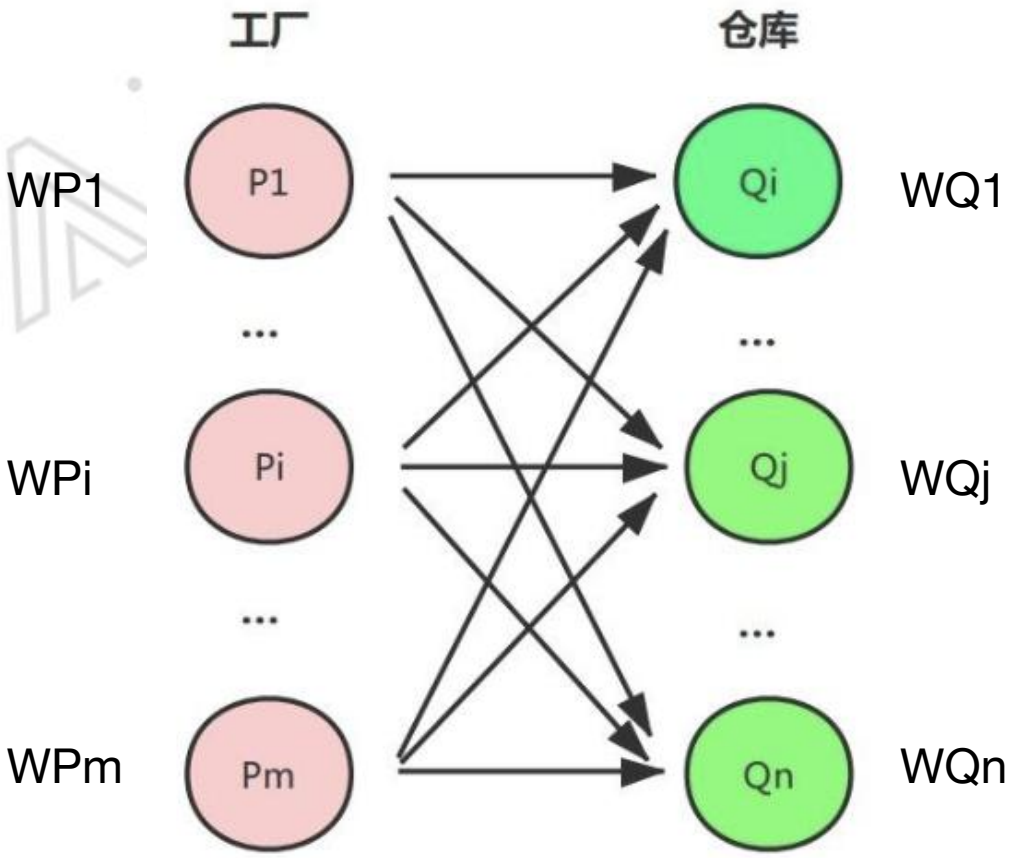


运输问题：从多个工厂运输货物到多个仓库。

- P从在P1 到 Pm代表m座工厂，工厂Pi有重量为Wpi的货物。
- Q从Q1到Qn代表n个仓库，仓库Qj最大容量为WQj。
- 每个仓库都希望装尽可能多的货物。如何尽可能高效地把所有货物从P运送到Q，就是运输问题的优化目标。

NeuHub AI Links All

EMD

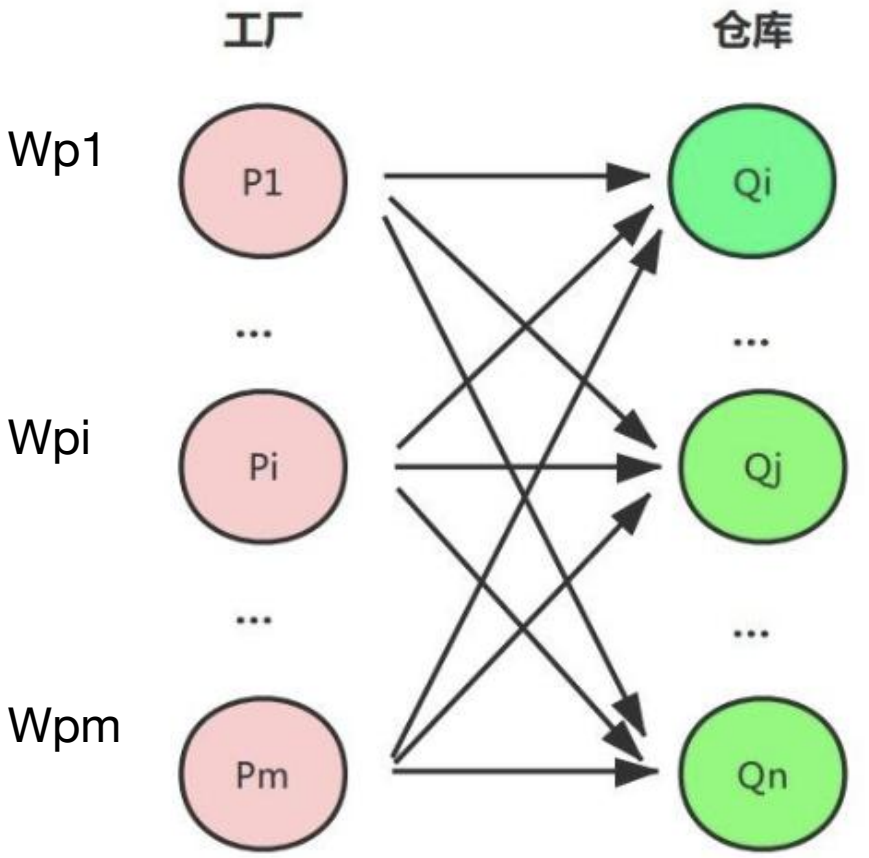


- 定义：货物从工厂Pi运到仓库Qj，距离是dij，运送货物的重量为fij，这样一次运输需要的工作量为dij*fij。
- 显然，距离越远或货物越重，工作量就越大。
- 工作量总和为W：

$$W = \sum_i^m \sum_j^n dij * fij \; -> \; min$$

NeuHub AI Links All

EMD

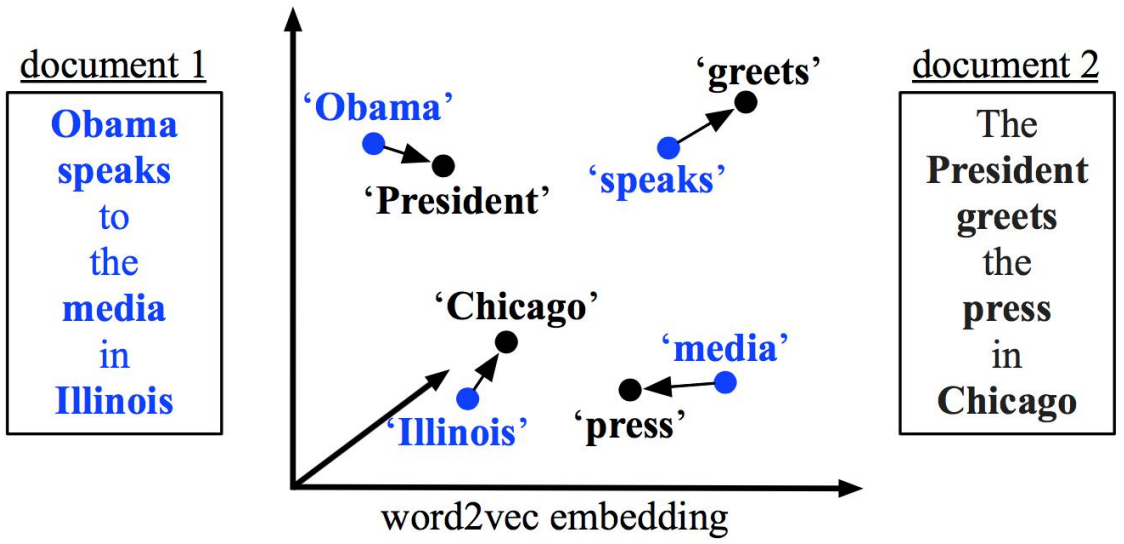


$$W = \sum_i^m \sum_j^n dij * fij \quad -> \ min$$

- 距离是确定的，所以运输量fij是式中唯一的变量，现有如下4个约束：
  - 运输过程从工厂P到仓库Q，不能反向 (fij >= 0)
  - 从工厂Pi一次次运出去的所有货物重量的和不可能超过该工厂中所有货物的总重量WPi ($\sum_{j=1}^{n} fij <= WPi$)
  - 仓库Qj接收的所有货物总重量不可能超过其总容量WQj ($\sum_{i=1}^{m} fij <= WQj$)
  - 总运输量的上限是工厂中货物总重、仓库总容量中的最小值
  - $\sum_{i=1}^{m}\sum_{j=1}^{n} fij = min(\sum_i^m WPi, \sum_j^n WQj)$
- 最终需要求最优解fij*

NeuHub AI Lifelong

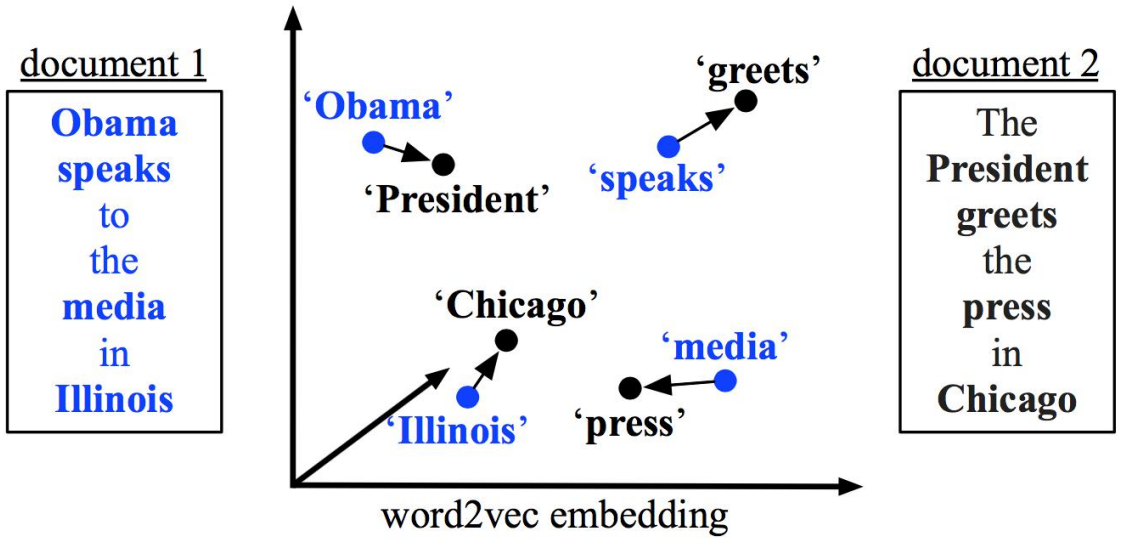
EMD



WMD与运输问题很相似：

- 可看成文档1转化为文档2所需要付出的总代价。
- 词与词之间的移动代价由语义距离（dij）决定，而权重矩阵（fij）则是最优化所需要求解的值。

文本相似度计算：WMD



naive的方式：

- 这两句话去除了停用词之后，分别有四个单词，每个单词用词向量进行表示。
- 句子之间距离：计算一下两个句子间最近的单词之间的距离，就可以快速知道两个句子间的差距了，对吗？

文本相似度计算：WMD

给定三个去除停用词后的短文本：
d1: 我/在/造/原子弹
d2: 原子弹/氢弹 /气泡弹
d3: 原子弹/西瓜/苹果

基于naive的计算方法：d1->d2 = d1->d3
事实上：d1->d2 < d1->d3

naive方法只考虑一一对应关系，而没有考虑到词与词之间多多对应的关系。

两份文本中每一个词都需要进行一一对应，计算一下相似度，并且是有侧重地计算相似度。

权重如何分配（如何侧重），需要满足什么条件？

NeuHub AI Links All

文本相似度计算：WMD

c(i,j)表示单词ij之间的相似度计算，而Tij表示对于这种相似度的权重。
由于两个文本之间的单词两两之间都要进行相似度计算，可知T是一个矩阵。
这个T要满足什么条件呢？

$$\min_{\mathbf{T}\geq 0} \sum_{i,j=1}^{n} \mathbf{T}_{ij} c(i,j)$$

$$\text{subject to: } \sum_{j=1}^{n} \mathbf{T}_{ij} = d_i \quad \forall i \in \{1,\ldots,n\} \qquad (1)$$

$$\sum_{i=1}^{n} \mathbf{T}_{ij} = d'_j \quad \forall j \in \{1,\ldots,n\}.$$

文本相似度计算：WMD

$$\min_{\mathbf{T}\geq 0} \sum_{i,j=1}^{n} \mathbf{T}_{ij} c(i,j)$$

$$\text{subject to:} \sum_{j=1}^{n} \mathbf{T}_{ij} = d_i \quad \forall i \in \{1,\dots,n\} \qquad (1)$$

$$\sum_{i=1}^{n} \mathbf{T}_{ij} = d'_j \quad \forall j \in \{1,\dots,n\}.$$

d是什么意思？
normalized BOW：单词的重要程度与词的出现频率相关（并且归一化）。

$$d_i = \frac{c_i}{\sum_{j=1}^{n} c_j}.$$

例子：你/站在/桥上/看/风景/看/风景/的/人/在/楼上/看/你
[2, 1, 1, 3, 2, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0]
[2, 1, 1, 3, 2, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0]/15

文本相似度计算：WMD

$$\min_{\mathbf{T}\geq 0} \sum_{i,j=1}^{n} \mathbf{T}_{ij} c(i,j)$$

$$\text{subject to:} \sum_{j=1}^{n} \mathbf{T}_{ij} = d_i \quad \forall i \in \{1,\dots,n\} \qquad (1)$$

$$\sum_{i=1}^{n} \mathbf{T}_{ij} = d'_j \quad \forall j \in \{1,\dots,n\}.$$

“你 ” 对应 2/15的权重，那么这个词分配到另一个文档d'中的词的所有权重加起来要等于2/15。假如另一个文档中总共有4个词：
你 -> d'w1 (1/30)
你 -> d'w2 (1/15)
你 -> d'w3 (1/60)
你 -> d'w4 (1/60)

同样对于d'，也需要对每个词计算nBOW，而 d'中某个词对应的被分配过来的权重加起来要等于这个词在d'文档中的权重。

文本相似度计算：WMD

目前开源的WMD实现有：

- https://pypi.org/project/wmd/
- https://github.com/mkusner/wmd

NeuHub
AI Links All

文本相似度计算：WMD

WMD（Word Mover's Distance）是2015年提出的一种衡量文本相似度的方法。优点：

- 效果出色：充分利用了word2vec的领域迁移能力
- 无监督：不依赖标注数据，没有冷启动问题
- 模型简单：仅需要词向量的结果作为输入，没有任何超参数
- 可解释性：将问题转化成线性规划，有全局最优解
- 灵活性：可以人为干预词的重要性

NeuHub
AI Links All

文本相似度计算：WMD

缺点：

- 词袋模型，没有保留语序信息

WMD认为『屡战屡败』和『屡败屡战』在语义上是完全一致的。

- 不能很好地处理词向量的OOV问题

由于词向量是离线训练的，应用于在线业务时会遇到OOV问题，用户query分出来的词，有可能找不到对应的词向量。

- 处理否定词能力偏差

在训练好的词向量中，通常语义相反的词的词向量是比较相近的，这会导致语义相反的两个句子WMD距离很近。

其实这也是其它很多相似度计算方法的缺点

NeuHub AI Links All

检索式对话系统的关键技术

京东智联云 & 贪心科技 GREEDY TECHNOLOGY

- 如何快速计算

NeuHub AI Links All

如何快速计算

粗排：

在海量数据中，一般要先经过一个“粗筛”的模块召回若干候选项，减少q-r匹配的工作量。

这个模块一般将用户query与语料库里query进行快速匹配，得到几十上百个候选回复。

NeuHub
AI Links All

如何快速计算

- 轻量级文本表征：BOW，TFIDF
- 编辑距离
- 倒排列表
- BM25

NeuHub
AI Links All

如何快速计算

- 倒排列表

单词ID	单词	倒排列表（DocID）
1	谷歌	1,2,3,4,5
2	地图	1,2,3,4,5
3	之父	1,2,4,5
4	跳槽	1,4
5	Facebook	1,2,3,4,5
6	加盟	2,3,5
7	创始人	3
8	拉斯	3,5
9	离开	3
10	与	4
11	Wave	4
12	项目	4
13	取消	4
14	有关	4
15	社交	5
16	网站	5

如何快速计算

BM25

bm25 是一种用来评价搜索词和文档之间相关性的算法，它是一种基于概率检索模型提出的算法：有一个query和一批文档Ds，现在要计算query和每篇文档D之间的相关性，先对query进行切分，得到单词qi，然后单词的分数由3部分组成：

- 每个单词的权重
- 相关性分数R：单词和D之间的相关性，单词和query之间的相关性

最后对于每个单词的分数求和，得到query和文档之间的分数。

$$Score(Q,d)=\sum_{i}^{n}W_iR(q_i,d)$$

NeuHub AI Links All

如何快速计算

京东智联云 & 贪心科技

- 每个单词的权重

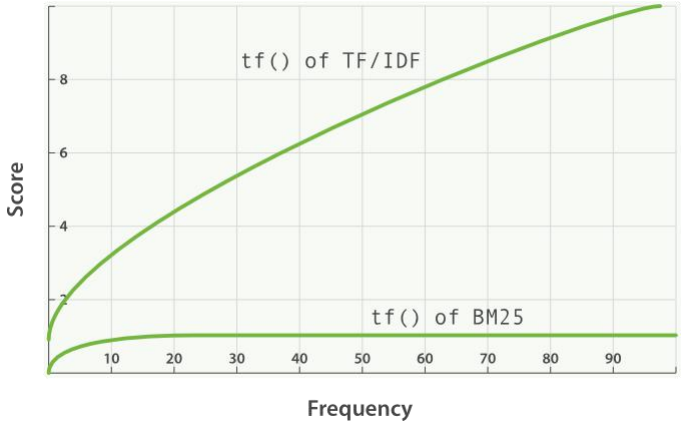
N表示所有文档数目，n(qi)为包含了单词qi的文档数目。

$$IDF(q_i) = log(\frac{N - n(q_i) + 0.5}{n(q_i) + 0.5})$$

NeuHub AI Links All

如何快速计算

- 单词和D之间的相关性，单词和query之间的相关性
- tf-idf中，用“词频”来代表与文本的相关性，但是词频和相关性并非线性，还和文档长度相关（较短文本更重要）。
- fi 是词项 t 在文档 d 中的词频，dl和 avgdl 分别是文档 d 的长度及整个文档集中文档的平均长度；k1，b 是参数（通常k1=1.2, b=0.75）



- qfi为qi在query中的词频；
- 简化情况下，k2一般取0。

$$R(q_i, d) = \frac{f_i \cdot (k_1 + 1)}{f_i + K} \cdot \frac{qf_i \cdot (k_2 + 1)}{qf_i + k_2}$$

$$K = k1 \cdot (1 - b + b \cdot \frac{dl}{avgdl})$$

检索式对话系统相关开源库：AnyQ

https://github.com/baidu/AnyQ

AnyQ系统集成了检索和匹配的众多插件，通过配置的方式生效；以检索方式和文本匹配相似度计算中的插件为例：

- 检索方式(Retrieval)
  - 倒排索引：基于开源倒排索引Solr，加入百度开源分词
  - 语义检索：基于SimNet语义表示，使用ANNOY进行ANN检索
  - 人工干预：通过提供精准答案，控制输出
- 匹配计算(Matching)
  - 字面匹配相似度：在对中文问题进行切词等处理之后，计算字面匹配特征
    - Cosine相似度
    - Jaccard相似度
    - BM25
  - 语义匹配相似度
    - SimNet语义匹配：使用语义匹配SimNet架构训练的模型，构建问题在语义层面的相似度

贪心科技 GREEDY TECHNOLOGY

京东智联云 & 贪心科技 GREEDY TECHNOLOGY

知识问答式对话系统的关键技术

NeuHub AI Links All

知识问答式对话系统的关键技术

- 知识图谱
- 问题解析、查询及答案生成

NeuHub
AI Links All

知识问答式对话系统的关键技术

基于医疗知识图谱的问答系统

用户:乳腺癌的症状有哪些?
小勇： 乳腺癌的症状包括：乳腺癌的远处转移；胸痛；乳头溢液；乳头破碎；肝肿大；泌乳障碍；乳头内陷；乳
****
用户:最近老流鼻涕怎么办?
小勇： 症状流鼻涕可能染上的疾病有：枯草热；副流行性感冒；急性上呼吸道感染；硫化氢中毒；小儿衣原体肺
****
用户:为什么有的人会失眠?
小勇： 失眠可能的成因有：躯体疾病和服用药物可以影响睡眠，如消化不良，头痛，背痛，关节炎，心脏病，糖
由于生活方式引起睡眠问题也很常见，如饮用咖啡或茶叶，晚间饮酒，睡前进食或晚饭较晚造成满腹食物尚未消
可能的原因有压力很大，过度忧虑，紧张或焦虑，悲伤或抑郁，生气，容易出现睡眠问题。
吵闹的睡眠环境，睡眠环境过于明亮，污染，过度拥挤。
****
用户:失眠有哪些并发症?
小勇： 失眠的症状包括：心肾不交；神经性耳鸣；咽鼓管异常开放症；偏执狂；十二指肠胃反流及胆汁反流性胃
****
用户:失眠的人不要吃啥?
小勇： 失眠忌食的食物包括有：油条；河蚌；猪油（板油）；淡菜(鲜)
****

https://github.com/vivianLL/QASystemOnHepatopathyKG

知识问答式对话系统的关键技术

&

第一步：需求是什么？

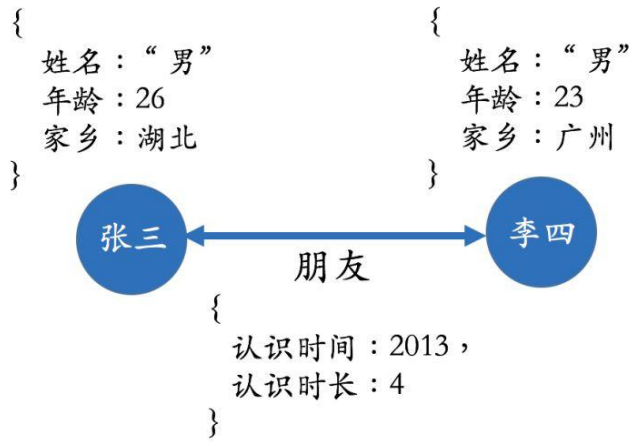
针对哪些用户？
使用场景？
可使用的数据资源？

NeuHub
AI Links All

知识问答式对话系统的关键技术

第二步：知识图谱的构建

业务需求：是否真的有必要构建知识图谱？
构建方式：专家式or自动抽取式
存储方式：neo4j
原型定义：有哪些实体，哪些关系，哪些属性？实体与关系的角色如何分配？



AI Links All

知识问答式对话系统的关键技术

第二步：知识图谱的构建
医药领域知识图谱

Links All

知识问答式对话系统的关键技术

第二步：知识图谱的构建
医药领域知识图谱

实体类型	中文含义	实体数量	举例
Check	诊断检查项目	3,353	支气管造影;关节镜检查
Department	医疗科目	54	整形美容科;烧伤科
Disease	疾病	8,807	血栓闭塞性脉管炎;胸降主动脉动脉瘤
Drug	药品	3,828	京万红痔疮膏;布林佐胺滴眼液
Food	食物	4,870	番茄冲菜牛肉丸汤;竹笋炖羊肉
Producer	在售药品	17,201	通药制药青霉素V钾片;青阳醋酸地塞米松片
Symptom	疾病症状	5,998	乳腺组织肥厚;脑实质深部出血
Total	总计	44,111	约4.4万实体量级

AI Links All

知识问答式对话系统的关键技术

第二步：知识图谱的构建
医药领域知识图谱

实体关系类型	中文含义	关系数量	举例
belongs_to	属于	8,844	<妇科,属于,妇产科>
common_drug	疾病常用药品	14,649	<阳强,常用,甲磺酸酚妥拉明分散片>
do_eat	疾病宜吃食物	22,238	<胸椎骨折,宜吃,黑鱼>
drugs_of	药品在售药品	17,315	<青霉素V钾片,在售,通药制药青霉素V钾片>
need_check	疾病所需检查	39,422	<单侧肺气肿,所需检查,支气管造影>
no_eat	疾病忌吃食物	22,247	<唇病,忌吃,杏仁>
recommand_drug	疾病推荐药品	59,467	<混合痔,推荐用药,京万红痔疮膏>
recommand_eat	疾病推荐食谱	40,221	<鞘膜积液,推荐食谱,番茄冲菜牛肉丸汤>
has_symptom	疾病症状	5,998	<早期乳腺癌,疾病症状,乳腺组织肥厚>
acompany_with	疾病并发疾病	12,029	<下肢交通静脉瓣膜关闭不全,并发疾病,血栓闭塞性脉管炎>

知识问答式对话系统的关键技术

第二步：知识图谱的构建
医药领域知识图谱

属性类型	中文含义	举例
name	疾病名称	喘息样支气管炎
desc	疾病简介	又称哮喘性支气管炎...
cause	疾病病因	常见的有合胞病毒等...
prevent	预防措施	注意家族与患儿自身过敏史...
cure_lasttime	治疗周期	6-12个月
cure_way	治疗方式	"药物治疗","支持性治疗"
cured_prob	治愈概率	95%
easy_get	疾病易感人群	无特定的人群

知识问答式对话系统的关键技术

第三步：问答类型定义及问答模块设计

问句类型	中文含义	问句举例
disease_symptom	疾病症状	乳腺癌的症状有哪些?
symptom_disease	已知症状找可能疾病	最近老流鼻涕怎么办?
disease_cause	疾病病因	为什么有的人会失眠?
disease_acompany	疾病的并发症	失眠有哪些并发症?
disease_not_food	疾病需要忌口的食物	失眠的人不要吃啥?
disease_do_food	疾病建议吃什么食物	耳鸣了吃点啥?
food_not_disease	什么病最好不要吃某事物	哪些人最好不好吃蜂蜜?
food_do_disease	食物对什么病有好处	鹅肉有什么好处?
disease_drug	啥病要吃啥药	肝病要吃啥药?
drug_disease	药品能治啥病	板蓝根颗粒能治啥病?
disease_check	疾病需要做什么检查	脑膜炎怎么才能查出来?
check_disease	检查能查什么病	全血细胞计数能查出啥来?
disease_prevent	预防措施	怎样才能预防肾虚?
disease_lasttime	治疗周期	感冒要多久才能好?
disease_cureway	治疗方式	高血压要怎么治?
disease_cureprob	治愈概率	白血病能治好吗?

GREEDY TECHNOLOGY
AI Links All

知识问答式对话系统的关键技术

第三步：问答类型定义及问答模块设计

分类

序列标注
句法分析
规则

Cypher

问句输入

意图解析

关键信息解析

查询语句

强直需要吃什么药？

药物查询

疾病：强直 关系：药物

知识图谱查询

你好，对于强直性脊柱炎的用药一般有萘普生阿司匹林布洛芬泼尼松，你要一定要在医师的指导之下，一旦发现这个病情应该及时的进行治疗，以防引起心肌炎脊柱关节畸形等并发症。

自然语言答案

结构化答案

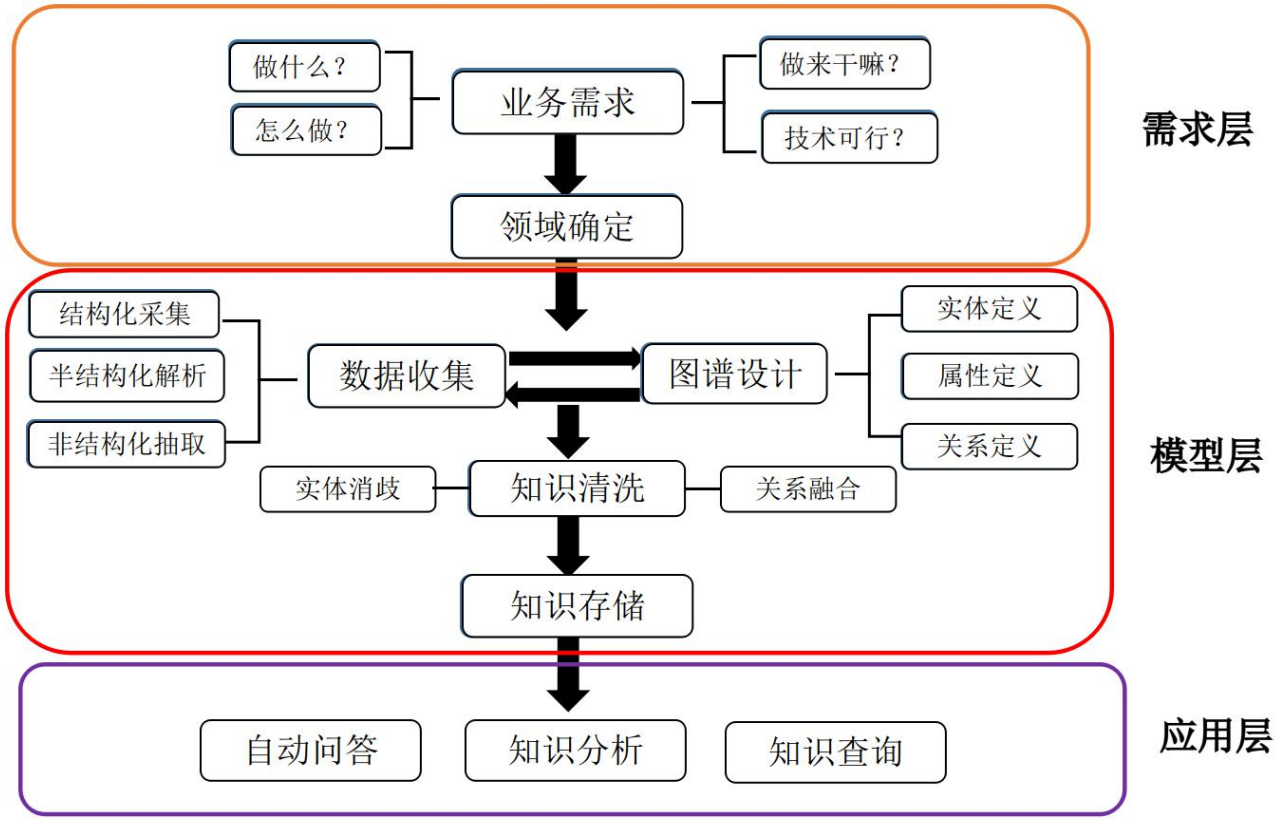
seq2seq
规则

萘普生阿司匹林布洛芬泼尼松

AI Links All

知识问答式对话系统的关键技术

总结



业务驱动的知识图谱构建框架

https://github.com/vivianLL/QASystemOnHepatopathyKG

贪心科技 GREEDY TECHNOLOGY

京东智联云 & 贪心科技 GREEDY TECHNOLOGY

任务式对话系统的关键技术

NeuHub AI Links All

任务式对话系统的关键技术

&

M: Hello, welcome to the Cambridge dialogue system, What kind of food would you like?
H: Yeah I want to find a restaurant that serves European food.
M: Hotel du Vin and Bistro is a nice place. It serves European food.
H: Uh what is the address and phone number?
M: Hotel du Vin and Bistro is a nice place. Their address is XXX and their phone is YYY.
H: Thank you goodbye.
M: Thank you for using this system.

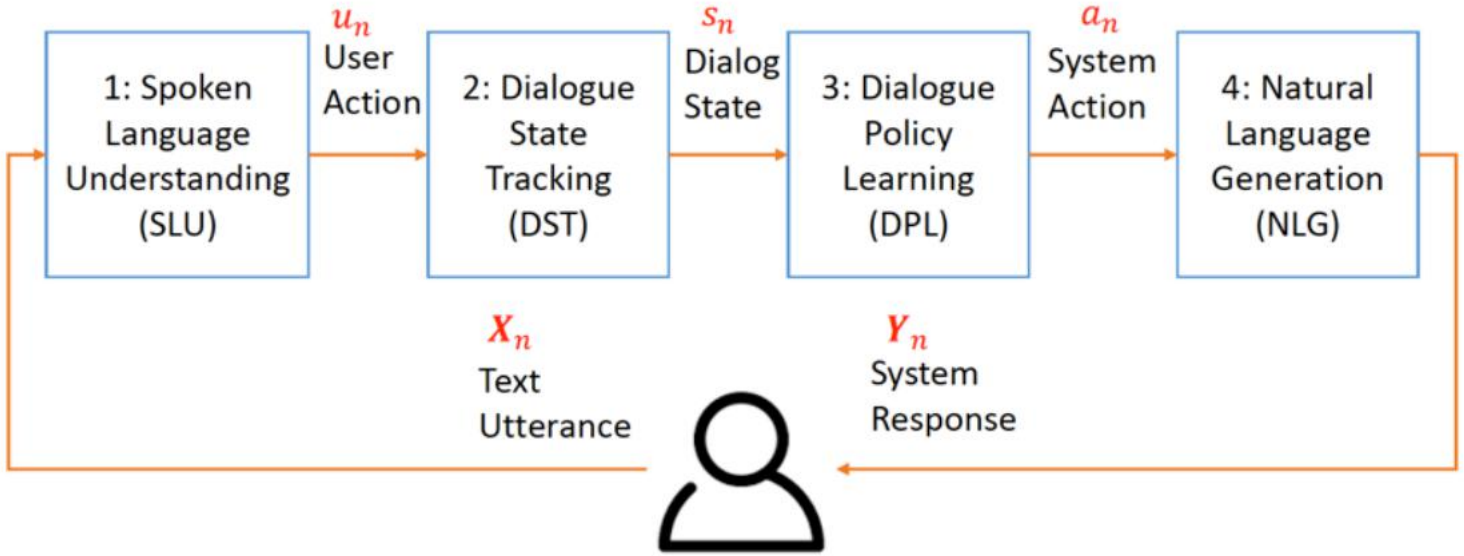
Task-oriented Domain-specific Dialog System

A: How old are you ?
B: I'm 16. Why are you asking ?
A: I thought you were 12.
B: What made you think so ?
A: I don't know what you are talking about.
B: You don't know what you are saying.

Chit-chat Open Domain Conversational Agent

Links All

任务式对话系统的关键技术

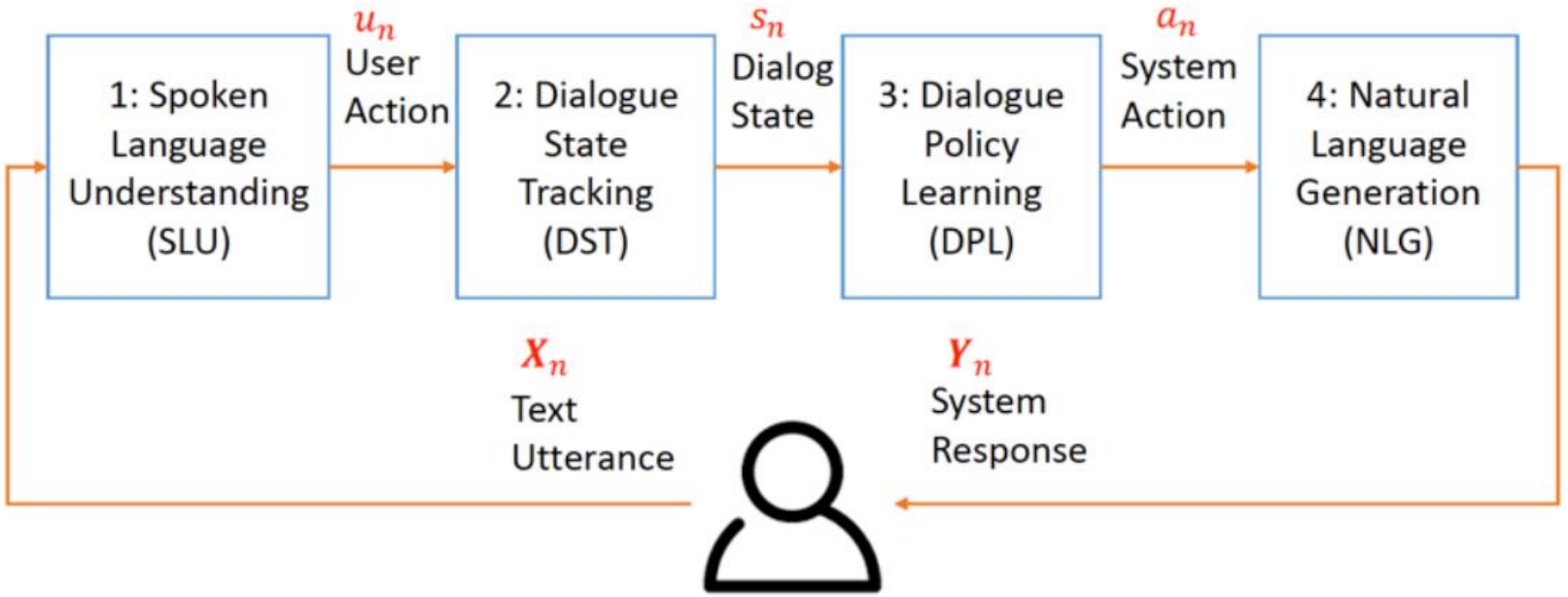


四大模块：

- 自然语言理解（Spoken Language Understanding，SLU）
- 对话状态追踪（Dialogue State Tracking，DST）
- 对话策略（Dialogue Policy Learning，DPL）
- 自然语言生成（Natural Language Generation，NLG）。

NeuHub AI Links All

任务式对话系统的关键技术



任务式对话系统的关键技术

&

- SLU模型的目标是识别用户的意图以及词槽填充（比如“我想吃西餐，价位便宜点的”，其意图是“找餐厅”，词槽是“菜系”和“价格”，词槽值分别是“西餐”和“便宜”）。
- DST的目的是保存并更新对话状态，包括一些历史对话信息、当前的词槽填充情况、数据库查询结果等等。
- DPL即基于当前的对话状态及用户输入作出合适的反馈。
- NLG则是将这个反馈转化为自然语言的过程。

NeuHub
AI Links All

任务式对话系统的关键技术

- 语义理解
- 语义追踪
- 强化学习的应用

NeuHub
AI Links All

任务式对话系统的关键技术

- pipeline系统，采用模块化结构：
  - 将四大模块分别进行独立处理。
  - 优点：这种模块化的系统可解释性强，易于落地，大部分业界的实用性任务型对话系统都采用的此结构。
  - 缺点：不够灵活，各个模块之间相对独立，难以联合调优，并且由于模块之间的误差会层层累积，单一模块的升级也可能需要整个系统一起调整。

NeuHub
AI Links All

任务式对话系统的关键技术

- 一个架构完备的多轮对话体系(pipeline)设计 需要考虑很多细节...

以槽的设计为例...

NeuHub
AI Links All

任务式对话系统的关键技术

槽（slot）：可分为必填与非必填

我：『去萧山机场多少钱』

出租车司机：『70』

对话中的『70』，应当被理解为70元人民币，而不必再去追问：『你说的是人民币、美元、日元还是港币?

NeuHub AI Links All

任务式对话系统的关键技术

词槽与接口槽

词槽：“利用用户话中关键词填写的槽”
接口槽：“利用用户画像以及其他场景信息填写的槽”

我（杭州用户）：『明天天气怎么样』

机器助手：『晴』

NeuHub
AI Links All

任务式对话系统的关键技术

同一个槽，可能会存在多种填槽方式，并且有优先级

我（杭州用户）：『我后天要坐火车去上海』

出发地：杭州?

出发地信息可来源于：用户所在地信息；用户日程表信息；用户主动提供

NeuHub AI Links All

任务式对话系统的关键技术

&

平级槽和依赖槽

订车票：需要知道用户出发的时间、地点、目的地、座位种类——这四个槽之间，没有任何依赖关系。

旅游规划：比如去海边和山里，每一个选择都会有影响到后续对话发展（海景房，山景房）也即 每个槽的填写结果会影响其它槽的填写。

NeuHub
AI Links All

任务式对话系统的关键技术

&

平级槽和依赖槽

如果错将平级槽采用依赖槽关系来管理，就会出现信息的丢失。比如 A、B、C，三者本为平级槽关系，但却将其用 A->B->C 的依赖槽关系来管理，那即便用户问句中包含填写 B、C 槽组的信息，也可能会由于 A 槽组的未填写而造成 B、C 槽组的填写失败。

如果错将依赖槽采用平级槽的关系来管理，就会出现信息的冗余，比如 A、B、C三者的关系为 A、A1->B、A2->C，那即便用户将值 A1 填入槽组 A 后，却仍然需要向用户询问本不需要的 C 槽组的填写信息。

NeuHub AI links All

任务式对话系统的关键技术

- 端到端系统
  - 训练一个从用户端自然语言输入到机器端自然语言输出的整体映射关系。
  - 灵活性强、可拓展性高，减少了设计过程中的人工成本，打破了传统模块之间的隔离。
  - 端到端模型对数据的数量和质量要求很高，并且对于填槽、API调用等过程的建模不够明确，现阶段业界应用效果有限，仍处在探索中。

NeuHub AI Links All

基于强化学习的任务式对话系统

&

End-to-End Task-Completion Neural Dialogue Systems

- 训练数据中的噪音
- 强化学习

NeuHub
AI Links All

基于强化学习的任务式对话系统优势

- 鲁棒性：在不确定性和混乱的情况下通过强化学习，自动选择动作，从而提出一种具有更高鲁棒性的神经对话系统。
- 灵活性：首个允许在对话过程中以用户主导的对话系统，用户可以在该系统中以更高的灵活性与系统进行交互，这在实际场景中很重要。
- 数据：应用规则模拟用户行为，因此不需要很多真实对话数据。

NeuHub
AI Links All

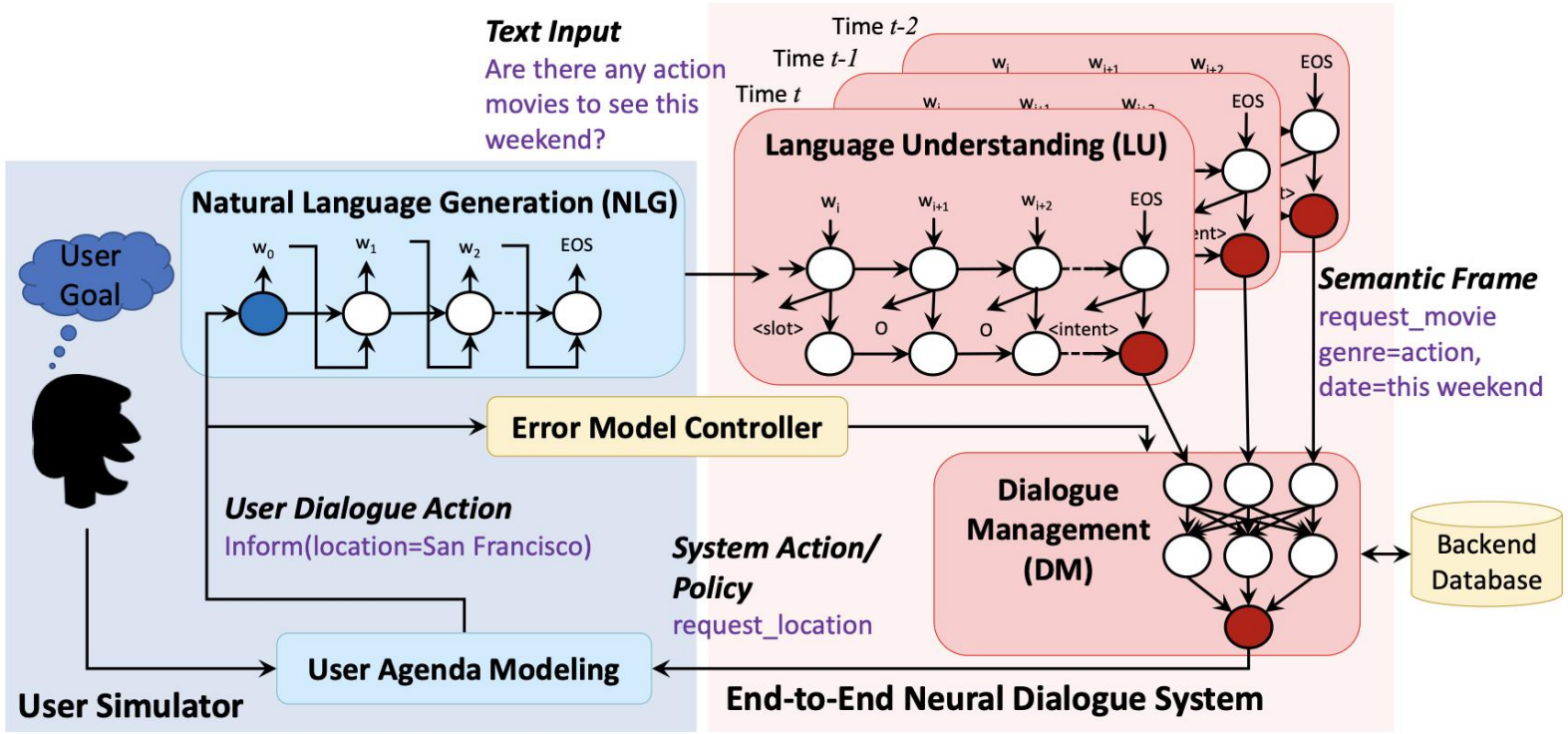
基于强化学习的任务式对话系统
基于规则VS基于RL

```
New episode, user goal:
{
    "request_slots": {
        "ticket": "UNK"
        "theater": "UNK"
        "starttime": "UNK"
    },
    "inform_slots": {
        "numberofpeople": "3",
        "date": "tomorrow",
        "moviename": "10 cloverfield lane"
    }
}
```

Rule-based Agent
User: What is the start time for 10 cloverfield lane? Agent: What movie are you interested in? User: I want to watch at 10 cloverfield lane. Agent: What time would you like to see it? User: What time is it playing tomorrow? I need 3 tickets. Agent: What city you would like? User: I do not care. Agent: What date would you like to watch it? User: I want to set it up tomorrow. Agent: Which theater would you like? User: Which theater is available? Agent: How many tickets do you need? User: I want 3 tickets please! Agent: Okay, I have booked 3 tickets for 10 cloverfield lane tomorrow. User: Thank you. Agent: Thank you.
Failure: Agent does not answer all the questions (starttime, theater) from user.

RL Agent
User: Which theater can I book 3 tickets for 10 cloverfield lane? Agent: What time would you like to see it? User: Which theater and start time are available tomorrow? Agent: 11:45am is available. User: Which theater is available? Agent: regal la live stadium 14 is available. User: Could you help me to book the tickets? Agent: Okay - I was able to book 3 tickets for you to see 10 cloverfield lane at regal la live stadium 14 at 11:45am tomorrow. User:Thank you. Agent: Thank you.
Success

Framework

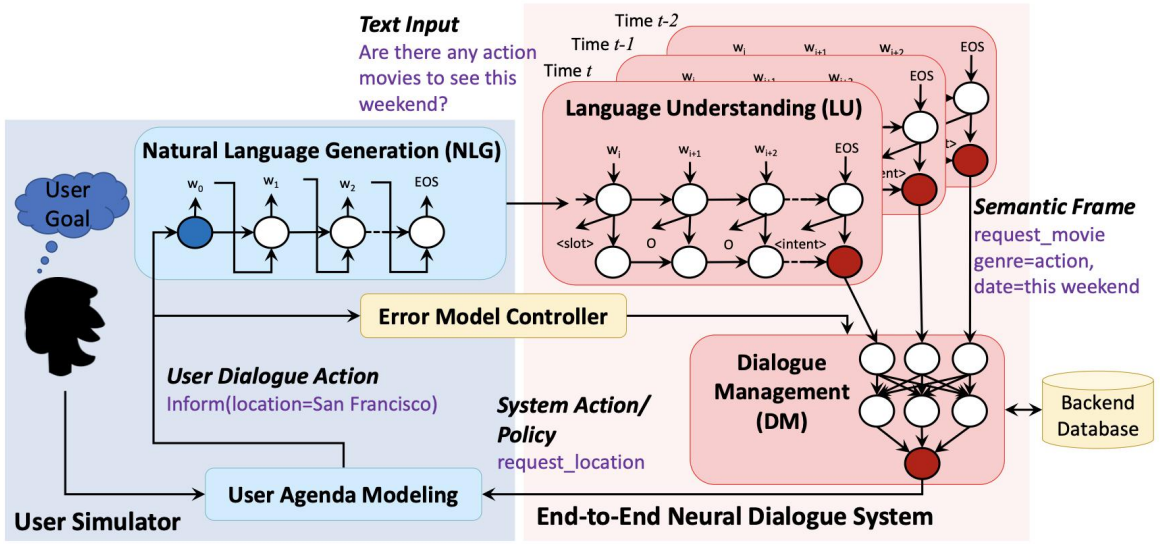


- 用户模拟器（左侧User Simulator）和基于强化学习的对话系统（右侧）。
- 在用户模拟器中，根据设定的User Goal，将基于神经网络将User Action转化为自然语言（NLG模块），以确保用户以上下文一致，面向目标的方式行事。
- 在神经对话系统中，输入语句通过LU模块并转化成为相应的语义框架（Semantic Frame），而DM（包括状态跟踪器和策略学习器）将累积每个语句的语义话语，在对话过程中稳健地跟踪对话状态，并生成下一个系统动作(System Action)。

Framework：User Simulator(用户模拟器)

A user goal:

- inform slots
- request slots



➢Template-based NLG:

每个时刻 t, user simulator 生成下一个user行为（基于当前状态以及最新agent 行为；更新状态）

➢Natural Language Generation (NLG)：

基于深度学习生成user行为

NeuHub AI Links All

Framework： Error Model Controller（错误控制器）

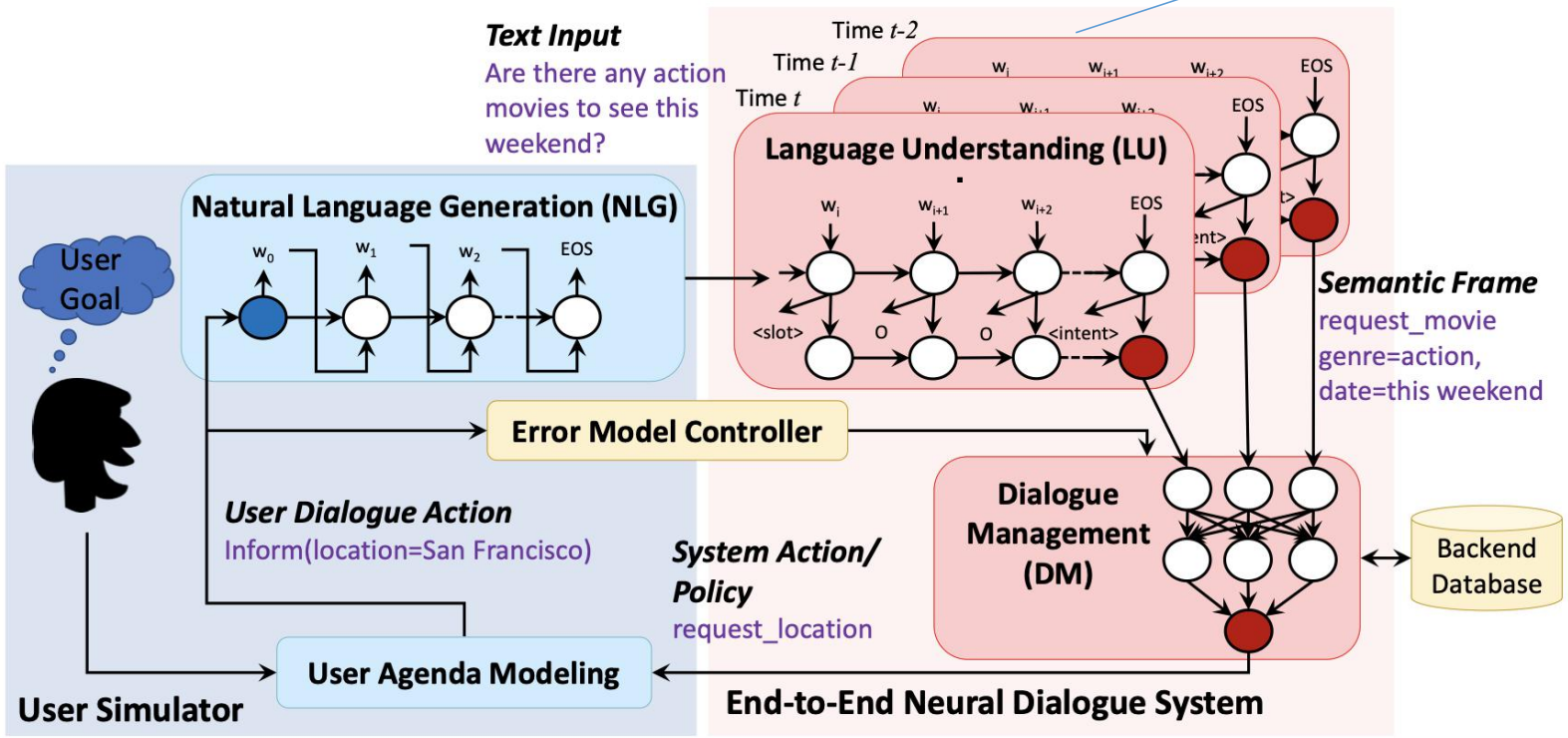
- **Action level error:**
  - ➢随机替换action的值
  - ➢替换整个slot
  - ➢删除整个slot
- **Intent level error:**
  - ➢随机替换intent

增强鲁棒性

NeuHub AI Links All

Framework：LU

W	find	action	movies	this	weekend
	↓	↓	↓	↓	↓
S	O	B-genre	O	B-date	I-date
I	find_movie				



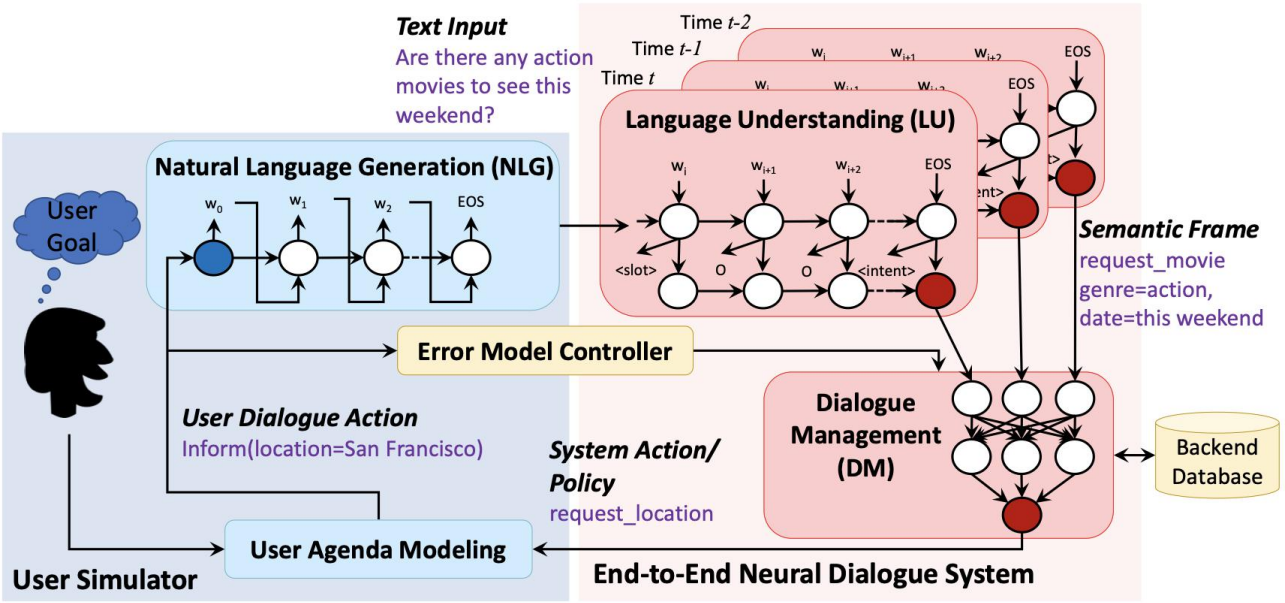
intent + slots

$$\vec{x} = w_1, ..., w_n, < EOS >$$
$$\vec{y} = s_1, ...s_n, i_m$$

$$\vec{y} = LSTM(\vec{x}).$$

$$p(\vec{y}|\vec{x}) = (\prod_{i}^{n} p(s_i|w_1, ..., w_i))p(i_m|\vec{y}).$$

Framework： DM



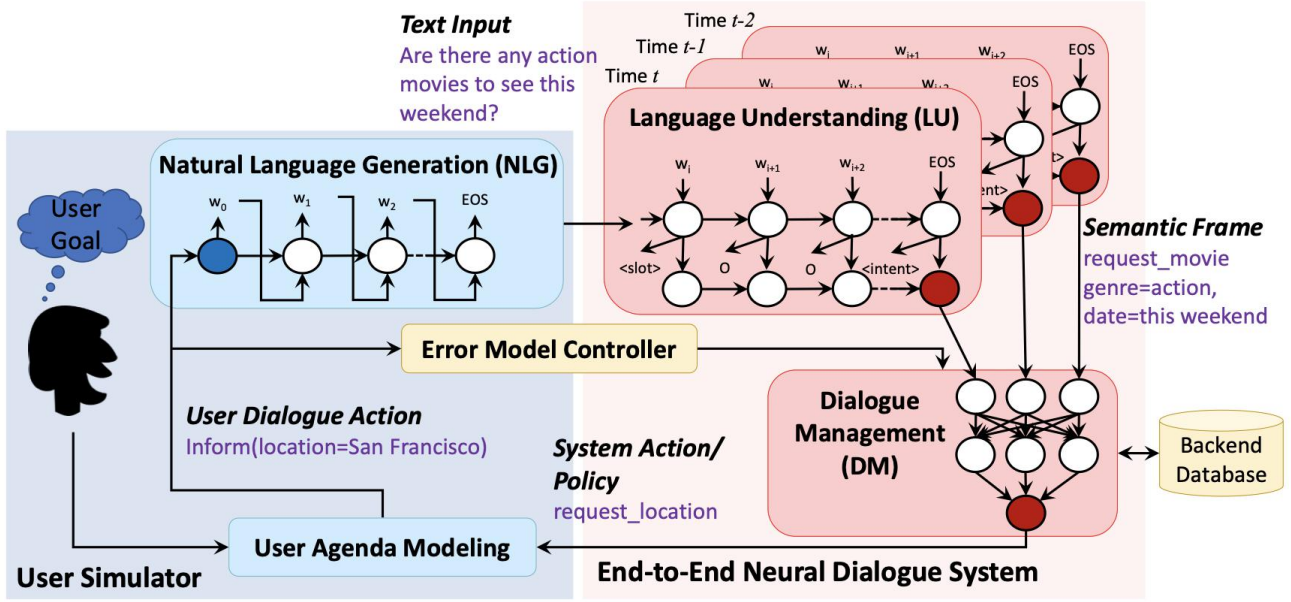
- Dialogue state tracking （对话状态跟踪）

➢输入(semantic frame): e.g., request(moviename; genre=action; date=this weekend)

➢数据库交互: 查询数据库获取信息

➢state tracker（状态追踪）：最新user行为(e.g., request(moviename; genre=action; date=this weekend)), 最新agent行为(request(location)), 数据库存查询信息, 回合信息, 历史回合信息, etc. -> 当前时刻的状态st

Framework： DM



- Policy learning （策略学习器）

policy π 生成agent action：π(st)

rule-based agent:

➢warm-start

➢trained end-to-end with RL

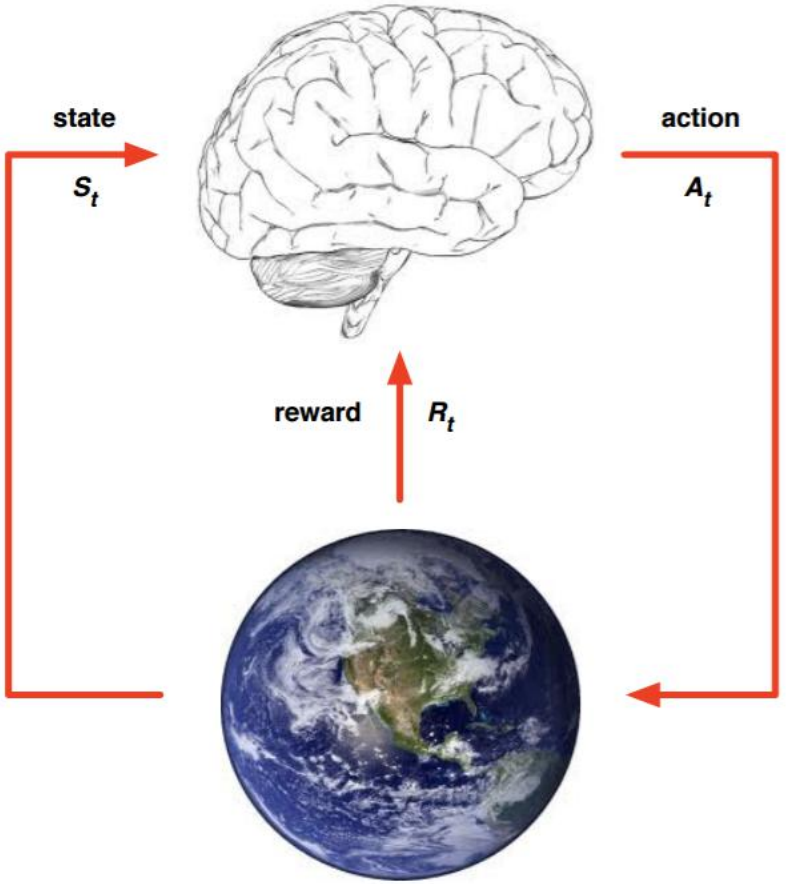
NeuHub AI LINE AI

Framework： End-to-End Reinforcement Learning

- The policy is represented as a deep Q-network (DQN) (Mnih et al., 2015), which takes the state st from the state tracker as input, and outputs Q(st, a; θ) for all actions a.
- 强化学习来和监督学习最大的区别是: 没有监督学习已经准备好的训练数据输出值。强化学习只有奖励值，和监督学习的输出值不一样，它不是事先给出的，而是延后给出的。
- 强化学习和非监督学习的区别也在于奖励值。

NeuHub AI Links All

Framework：End-to-End Reinforcement Learning

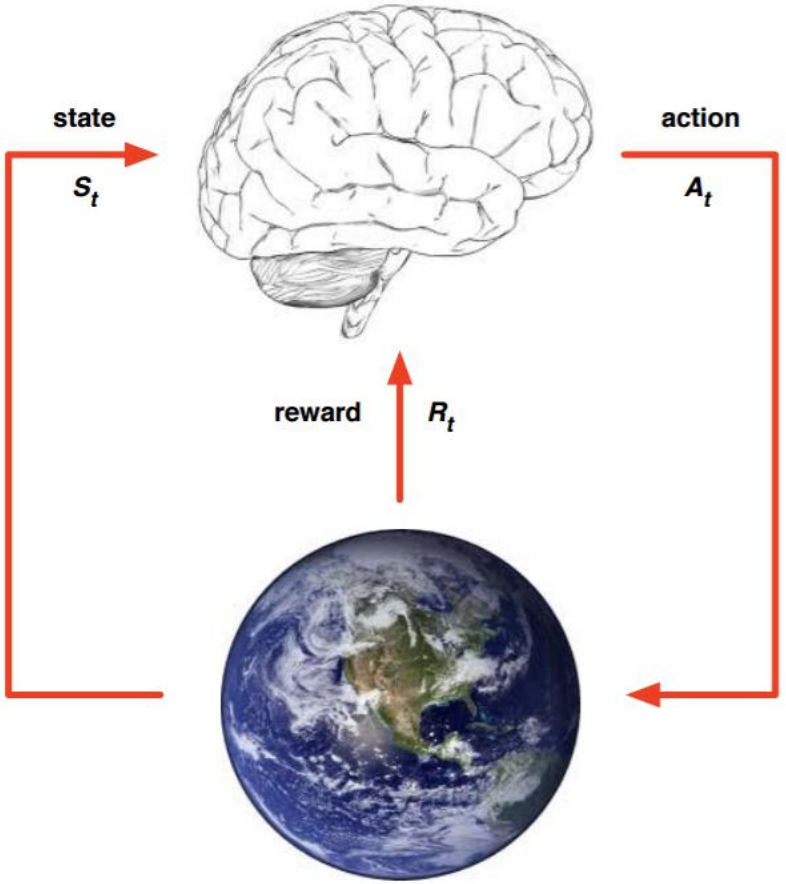


大脑代表算法执行个体，可以操作个体来做决策，即选择一个合适的动作（Action)At。

地球代表要研究的环境,有状态模型，在选择了动作At后，环境的状态(State)会变为St+1,同时得到所采取动作At的奖励(Reward)Rt+1。

NeuHub
AI Links All

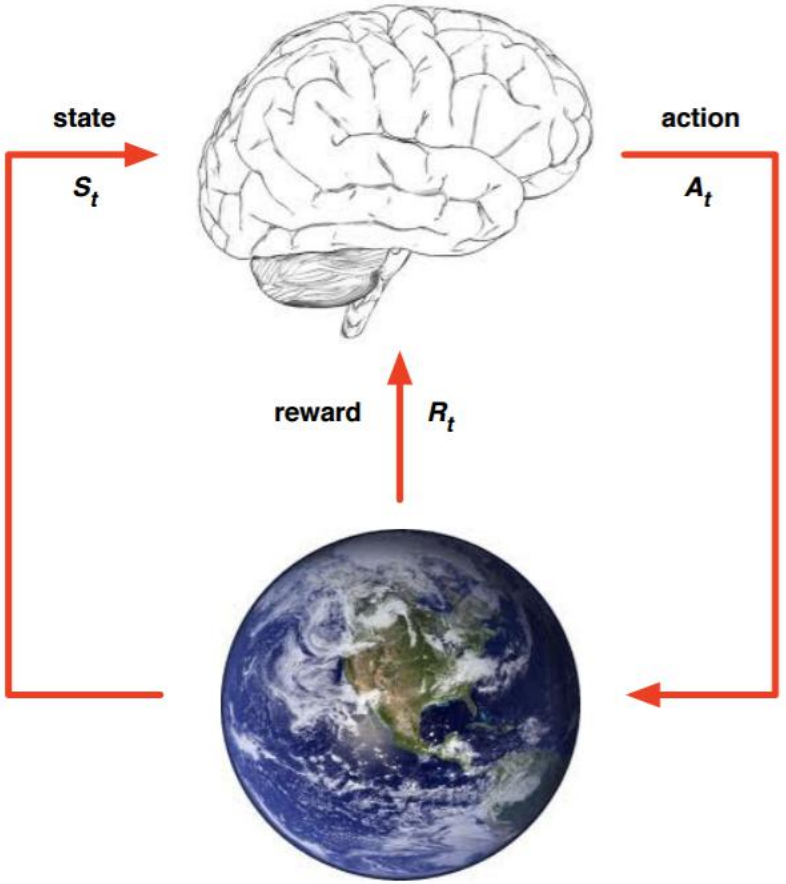
Framework：End-to-End Reinforcement Learning



- 环境的状态S, t时刻环境的状态称为St。
- 个体的动作A, t时刻个体采取的动作称为At。
- 环境的奖励R,t时刻个体在状态St采取的动作At对应的奖励Rt+1会在t+1时刻得到。

NeuHub AI Links All

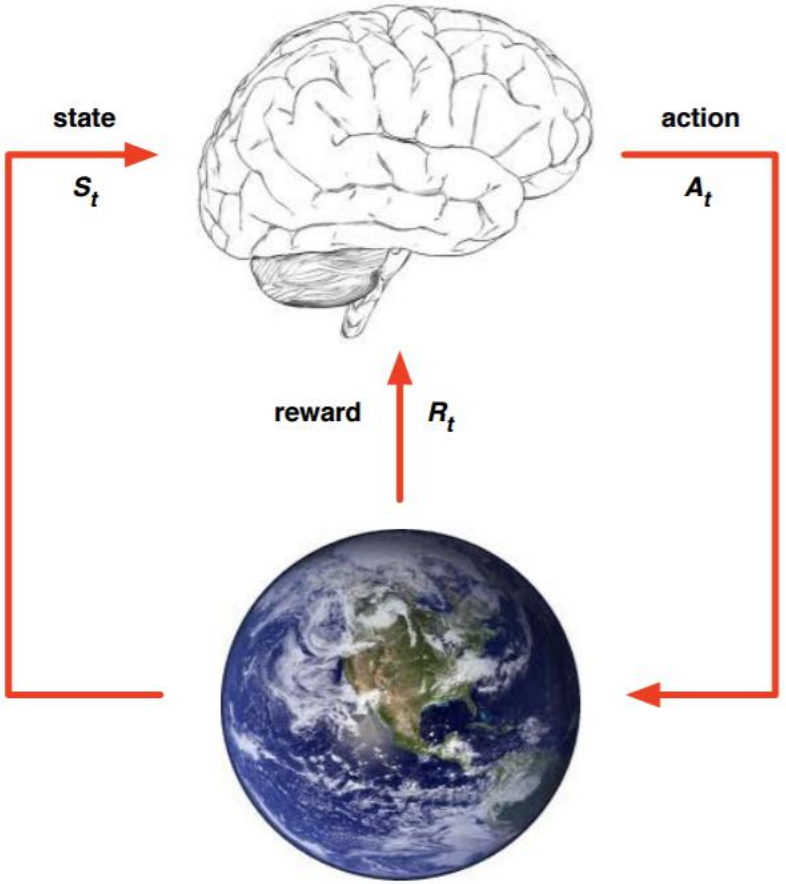
Framework：End-to-End Reinforcement Learning



- 个体的策略(policy)π,它代表个体采取动作的依据，即个体会依据策略π来选择动作。
- 个体在策略π和状态s时，采取行动后的价值（value），一般用Vπ(s)表示。这个价值一般是一个期望函数。
  Vπ(s)=Eπ(Rt+1+γRt+2+γ2Rt+3+...|St=s)
- γ即奖励衰减因子，在[0，1]之间。

NeuHub AI Links All

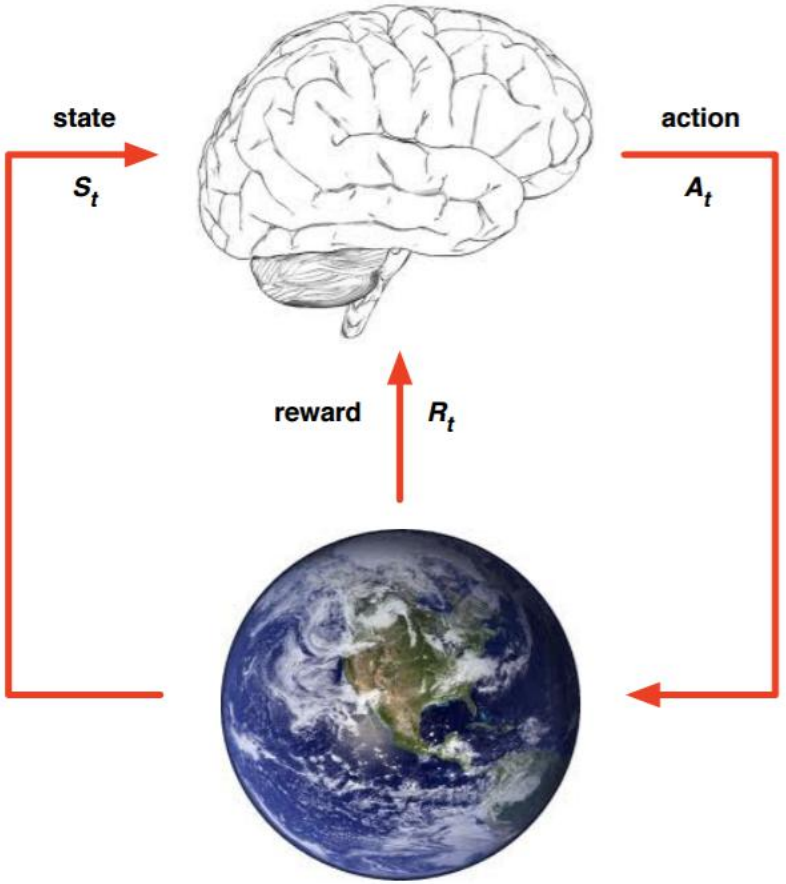
Framework：End-to-End Reinforcement Learning



- 环境的状态转化模型，可以理解为一个概率状态机，它可以表示为一个概率模型，即在状态s下采取动作a,转到下一个状态s′的概率，表示为Pss′a。
- 探索率ϵ，这个比率主要用在强化学习训练迭代过程中，由于我们一般会选择使当前轮迭代价值最大的动作，但是这会导致一些较好的但我们没有执行过的动作被错过。因此我们在训练选择最优动作时，会有一定的概率ϵ不选择使当前轮迭代价值最大的动作，而选择其他的动作。

京东智联云 &
NeuHub AI LinkSu

Framework：End-to-End Reinforcement Learning



DQN主要使用的技巧
经验回放（experience replay），即将每次和环境交互得到的奖励与状态更新情况都保存起来，用于后面目标Q值的更新。

NeuHub
AI Links All

Framework：End-to-End Reinforcement Learning

算法输入：迭代轮数T，状态特征维度n，动作集A，步长α，衰减因子γ，探索率ϵ，Q网络结构，批量梯度下降的样本数m。

输出：Q网络参数

1. 随机初始化Q网络的所有参数w，基于w初始化所有的状态和动作对应的价值Q。清空经验回放的集合D。

2. for i from 1 to T，进行迭代。

a) 初始化S为当前状态序列的第一个状态，拿到其特征向量$\phi(S)$

b) 在Q网络中使用$\phi(S)$作为输入，得到Q网络的所有动作对应的Q值输出。用$\epsilon-$贪婪法在当前Q值输出中选择对应的动作A

c) 在状态S执行当前动作A,得到新状态S'对应的特征向量$\phi(S')$和奖励R$,是否终止状态is_end

d) 将$\{\phi(S), A, R, \phi(S'), is_end\}$这个五元组存入经验回放集合$D$

e) $S = S'$

f) 从经验回放集合D中采样m个样本$\{\phi(S_j), A_j, R_j, \phi(S'_j), is_end_j\}, j = 1, 2., , , m$，计算当前目标Q值$y_j$：

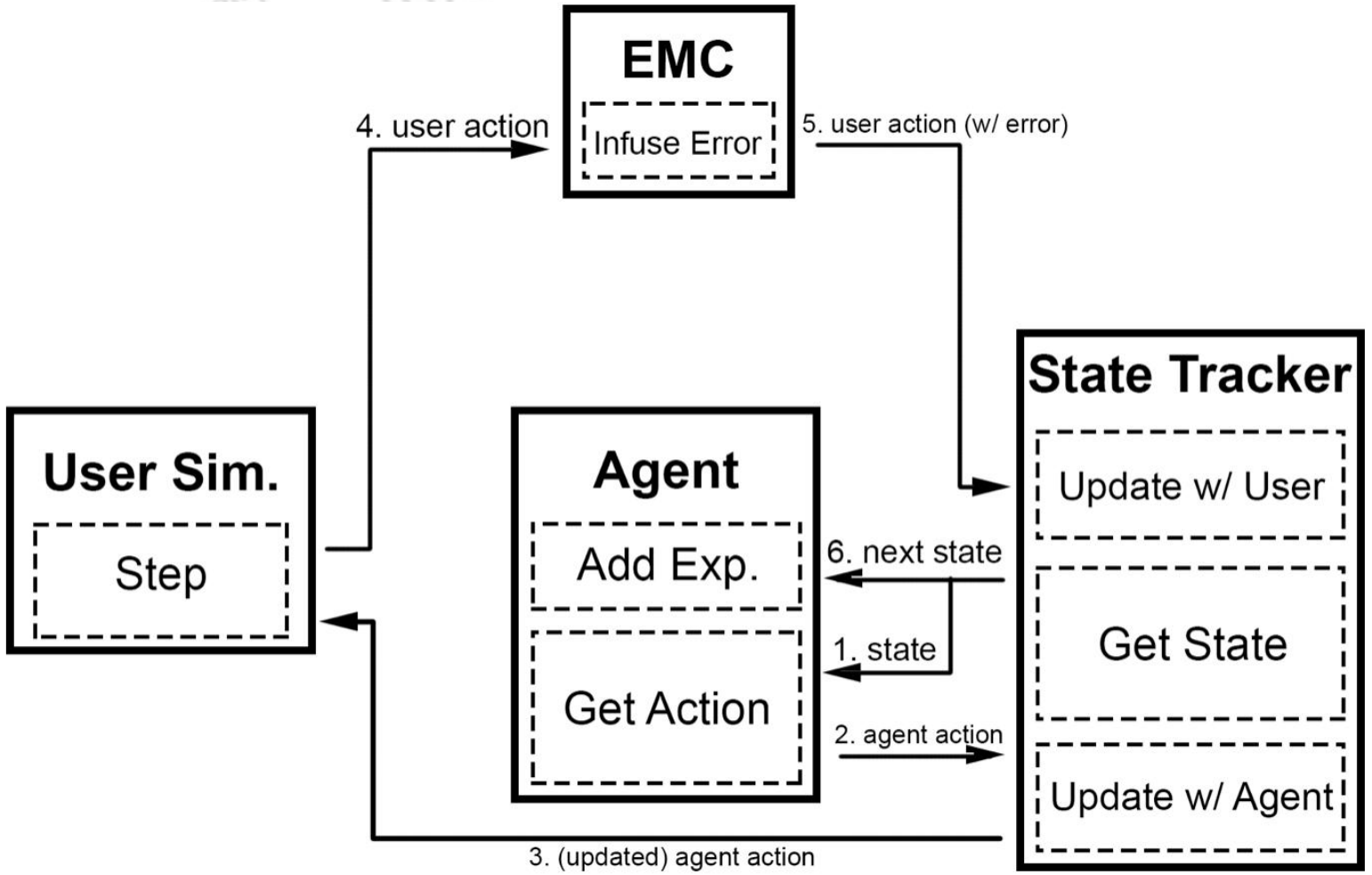
$$y_j = \begin{cases} R_j & is\_end_j\ is\ true \\ R_j + \gamma \max_{a'} Q(\phi(S'_j), A'_j, w) & is\_end_j\ is\ false \end{cases}$$

g) 使用均方差损失函数$\frac{1}{m}\sum_{j=1}^{m}(y_j - Q(\phi(S_j), A_j, w))^2$，通过神经网络的梯度反向传播来更新Q网络的所有参数w

h) 如果S'是终止状态，当前轮迭代完毕，否则转到步骤b)

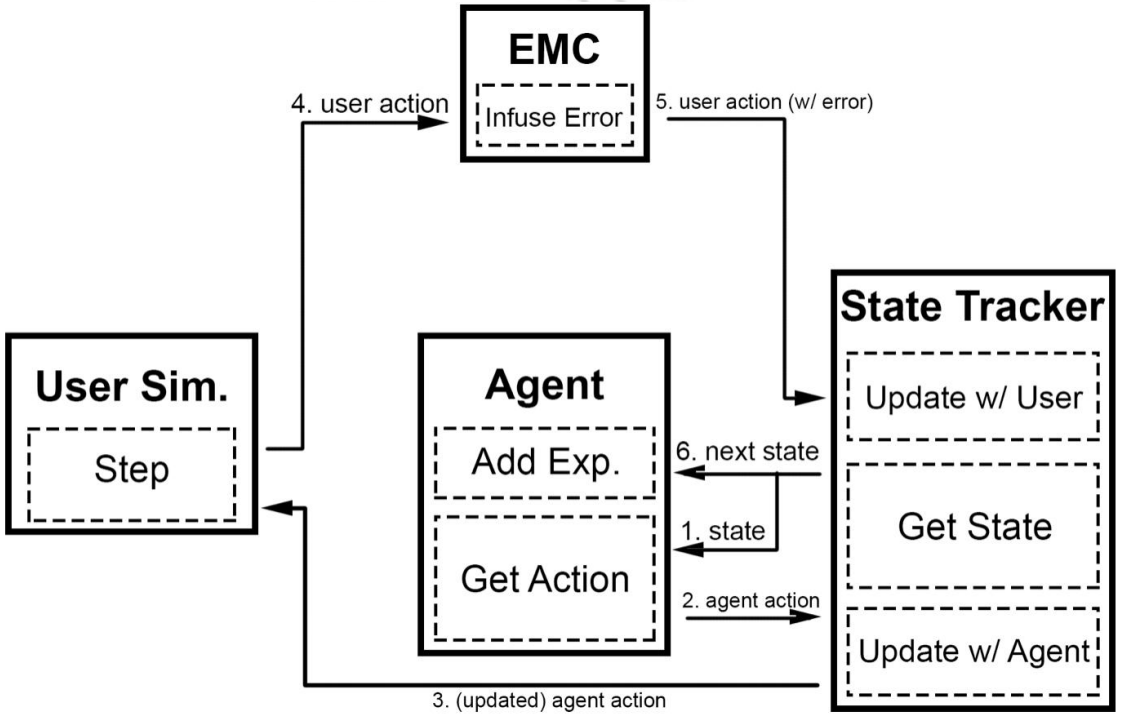
注意，上述第二步的f步和g步的Q值计算也都需要通过Q网络计算得到。另外，实际应用中，为了算法较好的收敛，探索率ϵ需要随着迭代的进行而变小。

Framework： process



Dialogue flow of a single round

Framework： process



Dialogue flow of a single round

1.Get the **current state** which is equivalent to the previous next state OR an initial state if this is the start of the episode and send it as input to the agent’s get action method

2.Get the **agent’s action** and send it to the ST

3.The **updated agent action** is sent as input into the user’s step method

4.The **user action** is infused with error by the EMC

5.The **user action** with error is sent as input into the ST

6.Finally, the next state is output from get state of the ST and this completes the current experience tuple of the round which is added to the **memory** of the agent

NeuHub AI Links All

参考链接

Emotional Chatting Machine: Emotional Conversation Generation with Internal and External Memory

https://www.sohu.com/a/136698085_500659

paper:
https://www.aclweb.org/anthology/I17-1074/

project code:
https://github.com/MiuLab/TC-Bot

project simplified code:
https://github.com/MiuLab/TC-Bot

project simplified medium blog:
https://towardsdatascience.com/training-a-goal-oriented-chatbot-with-deep-reinforcement-learning-part-i-introduction-and-dce3af21d383

NeuHub AI Links All